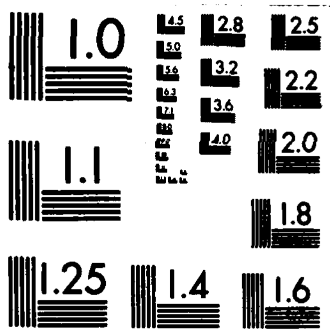
AD-A154 552 DEVELOPMENT OF REDUCED ORDER MODELS FOR CONTROL SYSTEM DESIGN USING THE OPTSYSX PROGRAM(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA S W NELSON DEC 84 1/2

UNCLASSIFIED F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

DEVELOPMENT OF REDUCED ORDER MODELS
FOR CONTROL SYSTEM DESIGN USING
THE OPTSYSX PROGRAM

by

Stanley William Nelson

December 1984

Thesis Advisor D.J. Collins

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER / 2. GOVT ACCESSION NO. AD-A154 552	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Development of Reduced Order Models for Control System Design Using the OPTSYSX Program	5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1984
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Stanley William Nelson	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943	12. REPORT DATE December 1984
	13. NUMBER OF PAGES 156
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Reduced Order Modelling
Model Reduction for Control Systems
F100 Turbofan Engine
X-29A Aircraft Control System

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The modern controls engineer is often faced with designing a system which is characterized by a large number of first order differential equations. It is highly desirable and sometimes necessary that such complex systems be reduced for analysis, synthesis and implementation into a physical system. It is the intent of this thesis to present a mathematical procedure and

and computer software based upon this procedure which enable the control engineer to formulate a reduced order model of a large order system.

As examples, two large order systems are analyzed: a sixteenth order model of the F100 turbofan engine and a ninety-eighth order model of the X-29A aircraft control system.

Approved for public release; distribution is unlimited

Development of Reduced Order Models for Control System Design Using the OPTSYSX Program

by

Stanley W. Nelson
Lieutenant Commander, United States Navy
B.S., University of Kansas, 1972

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1984

Author: Stanley W. Nelson
Stanley W. Nelson

Approved by: D. J. Collins
D.J. Collins, Thesis Advisor

H. A. Titus
H.A. Titus, Second Reader

M. F. Platzer
M.F. Platzer, Chairman,
Department of Aeronautics

J. N. Dyer
J.N. Dyer,
Dean of Science and Engineering

ABSTRACT

The modern controls engineer is often faced with designing a system which is characterized by a large number of first order differential equations. It is highly desirable and sometimes necessary that such complex systems be reduced for analysis, synthesis and implementation into a physical control system. It is the intent of this thesis to present a mathematical procedure and computer software based upon this procedure which enable the control engineer to construct reduced order models.

As examples, two large order systems are analyzed: a sixteenth order model of the F100 turbofan engine and a ninety-eighth order model of the X-29A aircraft control system. Additional keywords: OPTSYSX computer program; OPTRED computer program; interfaces; Fortran; computations; notch filters.

TABLE OF CCNTENTS

LIST OF FIGURES

SYMBOLS

A = State (Ns,Ns) or Output (No,No) Weighting Matrix
B = Control (Nc,Nc) Weighting Matrix
C = Control Gain Matrix (Nc,Ns)
D = Control (No,Nc) or Noise (No,Ng) Feedforward Matrix
Dr = Reduced Control Feedforward Matrix (No,Nc)
F = Open-Loop Dynamics Matrix (Ns,Ns)
Fr = Reduced Open-loop Dynamics Matrix (Nr,Nr)
G = Control Distribution Matrix (Ns,Nc)
Gr = Reduced Control Distribution Matrix (Nr,Nc)
GAM = State Disturbance Distribution Matrix (Ns,Ng)
H = Measurement Scaling Matrix (No,Ns)
Hr = Reduced Measurement Scaling Matrix (No,Nr)
K = Estimator Gain Matrix (Ns,No)
Nc = Number of Controls
Ng = Number of Process Noise Sources
Ns = Number of States
No = Number of Observations or Measurements
Nr = Reduced Model Number of States
Q = White Process Noise Covariance Matrix (Ng,Ng)
R = White Meas. Noise Covariance Matrix (No,No)
S = Steady-State Covariance Matrix of Control (Nc,Nc)
u = Control Vector (Nc,1)
v = White Measurement Noise Vector (No,1), With Zero Mean and Covariance Matrix R
w = White Process Noise Vector (Ng,1), with Zero Mean and Covariance Matrix Q
w0 = Constant Disturbance Vector (Ng,1)
x = State Vector (Ns,1)
$\dot{x}$ = Derivative of State Vector (Ns,1)

xe = Estimate of State Vector (Ns,1)

$\dot{xe}$ = Derivative of Estimate of State Vector (Ns,1)

y = Output Vector (No,1)

z = Measurement Vector (Nc,1)

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Professor D.J. Collins whose assistance and encouragement enabled this research. I would also like to thank Professor D.G. Williams, Director of the W.R. Church Computer Center, for his assistance and cooperation in providing the additional computer memory allocation which was essential to the completion of this work.

This thesis is dedicated to my wife Gayle, my daughter Kristin and my son Scott. Without their constant love, support and understanding this work would not have been possible.

I. INTRODUCTION

Modern control methods allow the control engineer to develop a control system to regulate every parameter of a physical system. He has been aided by computer simulations of non-linear systems and linear approximations of those systems. The desired control system can be developed parallel to or even ahead of the physical system which is to be controlled.

In many design applications areas, the engineer is confronted with designing a control system which can be represented analytically by a very large number of first order, linear, time invariant differential equations. This is especially true with highly complex and aero-elastic aircraft and space vehicles. The analytical model is a prime ingredient in the controller design process for any design technique.

Many practical limitations exist, such as computational requirements, that require the order (number of equations) of this complex model to be reduced for analysis, synthesis and actual implementation of the control system. Some method is required to analyze these dynamical models and establish simpler systems which include elements critical to the desired control function. Without such simplification, application of design procedures can result in highly comp nd parameter-sensitive controlled systems. With a reduced er model, regulator synthesis procedures become more intui e and far less sensitive to parameter variation. The educed order model is far easier to handle in formulating he control system but must characterize the physical system with sufficient accuracy such that performance objectives for the controlled physical system can be met by designing control laws from the reduced order model.

Both file and interactive data entry is accomplished by the main program and by the primary subroutine, REDUCX. The actual computation of the reduced order model is performed by subroutine REDUCX which utilizes subroutines MI and MAMULT for its matrix inversion and multiplication requirements. The remaining subroutines are utilized throughout OPTRED for interactive entry of numerical and character string data. The program requires no external subroutines for its execution.

The main program begins by presenting the user with a brief operational description of OPTRED. At this point, the user is given the option of having the program's general results repeated to the screen or printed to a listing file. OPTRED creates a permanent data file which contains the reduced data regardless of the user's option here. The program then reads full system flags and parameters, excluding matrices, from the OPTMAT data file. A brief description of the full system under investigation is then presented as an indication to the user that the desired data is being input. This description includes the order of the system, the number of controls and observations and whether or not a system "D" matrix will be input. The latter is indicated either by " " or "not" in the following example. The program then prompts the user for entry of the desired order of the reduced system.

The main program presentation as it appears during execution:

OPTRED

OPTRED WILL COMPUTE A REDUCED ORDER MODEL FROM FULL SYSTEM "F","G","H", AND "D" (IF INPUT) MATRICES.

THE FULL SYSTEM MATRICES MUST BE READ FROM A FILE NAMED "OPTMAT DATA" AS CREATED BY THE OPTSYSX PROGRAM.

YOU MUST ALSO ENTER THE DESIRED REDUCED ORDER (NUMBER OF STATES)

1) The compiled text file of OPTRED
2) The OPTMAT data file
3) The OPTRED EXEC file

The first two required files are self explanatory. The OPTRED EXEC file sets up the required read/write file definitions and calls the necessary library functions for execution using the IBM 3033 System. This EXEC allows the user to execute the OPTRED program by simply entering the word "OPTRED" at the terminal. The program then continues in a user-interactive mode until completion as described in the next section. The OPTRED EXEC file is listed in Appendix C.

B. PROGRAM OPERATION

OPTRED was written to satisfy two specific objectives. The first is to provide a method to assist in the analysis of actual large order control systems. The second is to create a program to be used as an instructional tool for students studying controls related problems.

To accomplish these objectives the emphasis while creating OPTRED was to make the program as user oriented as possible by eliminating data entry ambiguities and providing features which prevent the user from other inadvertant or invalid program entries. A serious effort was also made to minimize the amount of virtual memory required to execute this program.

1. Program Composition and Flow

OPTRED is composed of one main program and seven subroutines. The program can be divided into three basic catagories:

1) File Data Input
2) Interactive Data Input
3) Calculation

IV. THE OPTRED PROGRAM

A. PROGRAM OVERVIEW

OPTRED is an interactive, double precision FORTRAN program which uses a specifically formatted data file to input full system data. The required matrix inputs from this file are the full system "F", "G", "H" and "D" matrices. The system flags and parameters which OPTRED must also read for program execution are Ns, Nc, No, and IFDFW. The OPTSYSX program creates OPTMAT data which is the source of this required data. Through interactive input, the user specifies all other data which is required to create reduced model data.

OPTRED also reads from the OPTMAT file the noise, feedback control and the output and control cost weighting matrices. These are not required inputs but only input to the program to facilitate creation of a reduced order data file which is identical in format to the OPTMAT data file. The final result of OPTRED is the creation of OPTMATR, a data file which contains the reduced system "F", "G", "H" and "D" matrices and is formatted for re-entry into the OPTSYSX program.

It is possible for the user to construct a data file named OPTMAT to provide input data to OPTRED. However, only in rare applications would the full system not be analyzed prior to the development of the reduced model. OPTSYSX is therefore the simplest and most preferred method of data generation.

To execute the OPTRED program, a minimum of three separate files must be immediately accessible to the user:

$$Y = Hr*X_1 + Dr*U \tag{3.24}$$

where

$$Fr = F_{11} - F_{12}*F_{22}^{-1}*F_{21} \tag{3.25}$$

$$Gr = G_1 - F_{12}*F_{22}^{-1}*G_2 \tag{3.26}$$

$$Hr = H_1 - H_2*F_{22}^{-1}*F_{21} \tag{3.27}$$

$$Dr = D - H_2*F_{22}^{-1}*G_2 \tag{3.28}$$

This is the mathematical procedure which OPTRED uses in computing reduced order models. The complete OPTRED listing is included as Appendix B.

$$G' = \begin{bmatrix} G_1 \\ \hline G_2 \end{bmatrix} \tag{3.16}$$

$$H' = [H_1 \mid H_2] \tag{3.17}$$

If the states in X_1 truly model the full order system, then the subvector $\dot{X}_2$ will be essentially zero and

$$\dot{X}_1 = F_{11}*X_1 + F_{12}*X_2 + G_1*U \tag{3.18}$$

$$0 = F_{21}*X_1 + F_{22}*X_2 + G_2*U \tag{3.19}$$

Solving for X_2 in Equation 3.19, one has

$$X_2 = -F_{22}^{-1}*F_{21}*X_1 + F_{22}^{-1}*G_2*U \tag{3.20}$$

This is incorporated into Equation 3.18 and

$$\dot{X}_1 = (F_{11}-F_{12}*F_{22}^{-1}*F_{21})X_1 + (G_1-F_{12}*F_{22}^{-1}*G_2)U \tag{3.21}$$

and the output equation becomes

$$Y = (H_1-H_2*F_{22}^{-1}*F_{21})X_1 + (D-H_2*F_{22}^{-1}*G_2)U \tag{3.22}$$

The reduced order model can now be written as

$$\dot{X}_1 = Fr*X_1 + Gr*U \tag{3.23}$$

$$X' = R*X = T*Z' = \begin{bmatrix} X_1 \\ \hline X_2 \end{bmatrix} \tag{3.9}$$

This X' vector is the reordered state vector, where the elements contained in the subvector X_1 are those states associated with the designers choice of eigenvalues while the subvector X_2 contains all other states.

The linear system may now be rewritten

$$\dot{X}' = R*F*R^{-1}*X' + R*G*U \tag{3.10}$$

and

$$Y = H*R^{-1}*X' + D*U \tag{3.11}$$

Now define

$$F' = R*F*R^{-1} \tag{3.12}$$

$$G' = R*G \tag{3.13}$$

$$H' = H*R^{-1} \tag{3.14}$$

and

$$F' = \left[\begin{array}{c|c} F_{11} & F_{12} \\ \hline F_{21} & F_{22} \end{array}\right] \tag{3.15}$$

The eigenvalues which are to be included in the initial reduced order model are determined through modal analysis. The eigenvalue matrix can be re-ordered by constructing a matrix Q, which has the value 1 in the position corresponding to the eigenvalue to be included (column) and the re-ordered position (row) of that eigenvalue. For example, if eigenvalue 20 of the original Λ matrix was to be re-ordered to the first position, a 1 would be placed in the (1,20) position of the Q matrix. Now, the reordered matrix

$$\Lambda' = Q*\Lambda \tag{3.4}$$

is incorporated in the linear model

$$Q*\dot{Z} = \Lambda'*Z + Q*T^{-1}*G*U \tag{3.5}$$

letting

$$Z' = Q*Z \tag{3.6}$$

thus

$$\dot{Z}' = \Lambda'*Q^{-1}*Z' + Q*T^{-1}*G*U \tag{3.7}$$

and

$$X = T*Q^{-1}*Z' \tag{3.8}$$

A matrix R can be constructed such that

III. ANALYTICAL DEVELOPMENT OF A REDUCED ORDER MODEL

There are numerous methods of order reduction for control systems as discussed by Enns [Ref. 8] of Stanford University. This thesis uses the theory discussed by DeHoff and Hall [Ref. 9] which presents a method of reducing the order of the linear system based on dominant states chosen by the designer. This dominance is determined by modal analysis of the eigensystem under investigation. Dougherty [Ref. 10] utilized this same methodology in the development of a computational model used in the analysis of the F100 turbofan engine. A summary of this technique is provided below.

The method of reduction involves finding the eigenvector transform matrix, T, such that

$$X*T = T*\Lambda \tag{3.1}$$

where Λ is the diagonal eigenvalue matrix. Then, by defining an alternate state vector, Z, where

$$X = T*Z \tag{3.2}$$

the original system may be re-written in modal coordinates as

$$\dot{Z} = \Lambda*Z + T^{-1}*G*U \tag{3.3}$$

"F", "G", "H", and "GAM" matrices were complemented by the addition of routines which also save the "D" matrix. These new routines are analagous to existing "save" routines. In addition to these routines a new flag, ISAD, was added to the existing flags ISAF, ISAG, ISAH, and IGAM. This new flag provides the same function in "saving" matrices as do the existing flags. For program continuity, it was also necessary to add ISAD to subrcutine INNER, which is the driving subroutine in the OPTSYSX program.

Subroutines RDMATF, RDMAT, and WRTMAT also required minor changes to accomodate the "D" matrix. These changes consisted of the addition of "D" to the subroutine parameter lists and the addition of read or write statements for either re-entry of "D" into OPTSYSX or for writing "D" to the OPTMAT data file. The existing flag, IFDFW, was also added to the parameter lists of these routines. This flag indicates whether or not a "D" matrix will be input by the user. OPTRED uses this flag in its computational methods in the same manner as does OPTSYSX. Appendix A lists the OPTSYSX program as modified by this thesis.

additions to that program's size would cause user difficulty in obtaining sufficient memory for program execution. It is for this reason that Diel's OPTCALC, Laptas' OPTGRAPH and the program developed by this thesis have been developed as independent programs which depend upon various data created by OPTSYSX. The minor modifications this thesis introduces to the OPTSYSX program are not detrimental to its original capabilities.

B. OPTSYSX MODIFICATIONS

This thesis is confronted with the problem of analyzing very large control system models. Therefore, the additions of Hoden's subroutine SETUP and Diel's subroutines RDMATF, RDMAT, and WRTMAT were crucial to system analysis. These routines make multiple computer analysis runs possible without painstaking re-entry of system parameters and matrices. In particular, the system "F", "G", "H", and "Gam" are some of the matrices which are stored for re-entry into OPTSYSX or stored as data for use by other analysis programs. Subroutine WRTMAT is responsible for the storage of these matrices in the data file "OPTMAT". It is this file which OPTRED utilizes to input all pertinent full system data.

In many control applications, a control feed-forward matrix ("D") is present in the system model. Therefore, this thesis makes provisions within OPTSYSX which enable the user to save the "D" matrix in the same manner by which the other system matrices are saved. Subroutine SETUP, which enables the user to input system matrices from file data, was changed to enable the user to input the "D" matrix. This consisted of adding "D" to the routine's parameter list and also added a read statement to input that matrix from a data file. Present routines in OPYSYSX which "save" the

open loop transfer function

$$[H]*|s[I] - [F]|^{-1}*[G] \qquad (2.5)$$

closed loop noise transfer function

$$[H]*|s[I] - [F]|^{-1}*[Gam] \qquad (2.6)$$

compensator transfer function from measurement to input

$$[C]*|s[I]-[F]+[G]*[C]+[K]*[H]|^{-1}*[K] \qquad (2.7)$$

where

- x = state vector (Ns X 1)
- $\dot{x}$ = derivative of the state vector (Ns X 1)
- u = control vector (Nc X 1)
- y = output vector (No X 1)
- z = measurement vector (No X 1)
- w = white process noise vector (Ng X 1)
- [F] = open-loop dynamics or plant matrix (Ns X Ns)
- [G] = control distribution matrix (Ns X Nc)
- [Gam] = state disturbance distribution matrix (Ns X Ng)
- [H] = measurement distribution matrix (No X Ns)
- [D] = control feed-forward matrix (No X Nc)
- [C] = control gain matrix (Nc X Ns)
- [I] = identity matrix (Ns X Ns)

OPTSYSX is an extremely large and complex program which contains over 4000 lines of code. Its existing standard dimensions will accomodate a thirty-second order system with up to 10 controls and 10 observations or measurements. To execute even this version of OPTSYSX the user requires one megabyte of virtual storage. Any further significant

II. THE OPTSYSX COMPUTER PROGRAM

A. PROGRAM OVERVIEW

OPTSYSX is a double precision, interactive FORTRAN program which employs modern control theory analysis techniques. It is developed to be compiled and executed by the Naval Postgraduate School's IBM 3033 System 360/370. Its primary capabilities include the calculation of the open loop eigensystem, and the fixed closed loop system; the synthesis of optimal regulators or filters; the power spectral density, and modal distribution computations.

The fundamental system equations used by the OPTSYSX program for its computations are of the state variable form. The basic system equations are:

system model

$$\dot{x} = [F]*x + [G]*u + [GAM]*w \qquad (2.1)$$

output equation

$$y = [H]*x + [D]*u \qquad (2.2)$$

measurement equation

$$z = [H]*x + v \qquad (2.3)$$

estimator equation

$$\dot{xe} = [F]*xe + [G]*u + [K]*(z - [H]*xe) \qquad (2.4)$$

systems and the corresponding reduced models which OPTRED generates are presented and analyzed. Those systems include a sixteenth order model of the F100 turbofan engine and a ninety-eighth order model of the X-29A aircraft. Complete program listings for the OPTSYSX program and the OPTRED program are included as Appendices A and B respectively.

The intent of this thesis is to present the mathematical basis for creating such a reduced order model and to develop the actual computer software which creates that reduced model based on a given large order system. This computer program (OPTRED) is developed to interface with an existing control analysis FORTRAN program named OPTSYS. This thesis will discuss the modifications or additions to the OPTSYS program which are necessary to that interface.

The OPTimal SYStems control program was originally developed by Hall [Ref. 1] for the study and application of optimal systems control theory. Later modifications to the program were made by Walker [Ref. 2] and Liu [Ref. 3] of Stanford University. Hoden [Ref. 4] modified the OPTSYS program to present a user-friendly, interactive version of the program (OPTSYSX). Diel [Ref. 5] introduced changes which enabled the user to save various data for re-entry into the program and also to create data files which were necessary for the execution of a time response program (OPTCALC). Further modifications have been made by Laptas [Ref. 6] which were necessary to create data sets for input to his OPTGRAPH program. This FORTRAN program enabled the user to obtain Pole-Zero, Root-Locus, Nyquist, Bode and Nichols plots.

It is assumed that the user has a basic understanding of the fundamental concepts of control theory and optimal systems design. The symbology conventions of Bryson [Ref. 7] are used in the discussion of program operation and system descriptions.

An overview of the OPTSYSX program, its capabilities, and the modifications to that program which are necessary for the interface with OPTRED is presented first. This is followed by the mathematical basis upon which OPTRED is formulated and by a full description of the operation of that program. Finally, examples of large order control

AND THE ACTUAL STATE #'S (IN ASCENDING ORDER) WHICH REPRESENT THE REDUCED MODEL. THE ORDER OF THE REDUCED ORDER MODEL MUST BE LESS THAN THE ORDER OF THE FULL SYSTEM.

DO YOU WISH TO CONTINUE?

TYPE "YES" OR "NO".

DO YOU WISH RESULTS TO SCREEN OR DISK?

NOTE: A DATA FILE CONTAINING THE REDUCED ORDER DATA WILL BE GENERATED REGARDLESS OF YOUR ANSWER.

TYPE "S" FOR SCREEN OR "D" FOR DISK.

THE ORDER OF THE FULL SYSTEM IS:
THE NUMBER OF CONTROLS IS:
THE NUMBER OF OBSERVATIONS IS:
A "D" MATRIX WILL BE INPUT.

DO YOU WISH TO CONTINUE?
TYPE "YES" OR "NO".

ENTER THE DESIRED REDUCED ORDER OF THE "F" MATRIX.

Subroutine REDUCX is now called and immediately reads the full system matrices from file data. The user is then prompted for entry of the significant states which will compose his reduced model. This program makes a provision for entry of significant states from a specifically formatted, user created data file. At this point in the program he is given that option. Once the significant states have been entered, the program repeats those states to the screen and the user is given the option to make changes at that time. If the significant states are satisfactory to the user, the program continues with the computation of the reduced order model. At the completion of OPTRED, the reduced system data is created and can be exam-

ined for stability, controllability and observability by other analysis techniques as discussed in the following section. Prior to exiting the OPTRED program, the user is provided the option of re-executing OPTRED if he finds initial results unsatisfactory. A typical recording session which depicts this program flow is presented in the next section.

2. User Protection Features

The user of OPTRED is protected from abnormal program termination in several ways. Initially, the program presents its capabilities and user required entries. If he is not prepared to continue at this point, the user is offered the option of exiting the program. Next, the full system parameters are presented and the user is once again able to exit if these parameters are unexpected.

OPTRED requires that the order of the reduced system be greater than zero but less than the order of the full system. If the user inadvertantly enters a number which is out of this range, the program issues a warning to that effect and allows recovery. The computation of the reduced order model relies upon the construction of a reduced state matrix. OPTRED requires that these states be entered in the order that they would normally appear in this matrix. For this reason, the user must enter the significant state numbers in ascending order and those state numbers must lie in the range from one to the number of states in the full system. If the user inadvertantly enters these state numbers out of order or out of range the program prompts a warning and allows recovery.

The method of reduced order computation involves the inversion of the state submatrix F_{22}, as described in Chapter III, Equations 3.25 through 3.28. The nature of the plant matrix and the selection of reduced order states can

be coupled such that this matrix will be singular or non-invertable. OPTRED will detect this condition and issue information to the user that a reduced order model cannot be calculated for that full system from the given desired reduced states. Reduced data will still be computed but the user is warned that this data is invalid.

Subroutines RDINT, RDREAL and RDCHAR are responsible for the interactive input of data and expect an integer input, a real number input and a logical "YES" or "NO" input respectively. If the user inadvertantly makes an incompatible entry these subroutines issue warnings and allow another opportunity for data entry. The entry of a null line is included in these improper actions and the entry of two consecutive null strings will cause termination of the program. This function allows the user a further means of exiting OPTRED if he so desires.

3. Large Order Systems

The analysis of large crder systems, particularly the ninety-eigth order model, presents a major problem to the user with limited virtual memory assets. The longitudinal control system analyzed by this thesis has a (98 X 98) "F" matrix, a (2 X 98) "H" matrix and a (98 X 1) "G" matrix. To analyze this system, OPTSYSX must be executed in its "increased dimension" form. To compile and execute this version of OPTSYSX requires 2.5 Megabytes of virtual memory.

Both OPTRED and OPTGRAPH are also dimensioned to enable analysis of a ninety-eigth order model. When OPTSYSX, OPTRED, OPTGRAPH and their peripheral data are all used during a typical analysis session, very careful attention to file and virtual memory management is mandatory to prevent the user from exceeding his virtual machine's capacity.

C. INTERFACE WITH OTHER ANALYSIS PROGRAMS

To sufficiently analyze a control system it is desirable to utilize OPTSYSX, OPTCALC, CPTGRAPH and OPTRED. It may also be desirable to conduct this analysis during a single, continuous interactive computer session. However, in the analysis of very large systems this concept presents serious difficulties in the areas of program flow, data management and physical memory assets. For this reason, OPTRED has not been developed to be fully autcmatic in its interface with these other analysis programs.

The OPTRED program requires the existance of the OPTMAT data file. As previously discussed, OPTRED generates the OPTMATR data file which contains the reduced model data. To analyze this data using OPTSYSX, the user must now rename the OPTMATR data file for entry into OPTSYSX as OPTMAT data. Care must be exercised at this point to either erase the full system data file or to rename it to prevent ambiguous data files. Although this procedure may seen cumbersome at first, it is easily accomplished in the XEDIT mode and provides the user with the capatility of maintaining several system data files which are readily available for analysis.

After OPTSYSX processes the reduced data, additional data files are created which enable the execution of other system analysis programs. OPTGRAPH and OPTCALC utilize the OPTMAT and OPTGROL data files, respectively, to provide the functions described in chapter 2. When the user is investigating several systems it is both prudent and convenient to rename these files after their use for the purpose of future analysis.

D. EXAMPLES OF ORDER REDUCTION

Two large order systems will be presented to evaluate reduced models generated by OPTRED. The first system is a

sixteenth order model of the F100 turbofan engine. The final example is a ninety- eighth order model of the X-29A aircraft's longitudinal control system. Bode analysis was conducted for the full and reduced models and graphical comparisons follow at the end of the chapter. Recorded terminal sessions for each example will be presented to fully illustrate actual program operation.

1. Example of a Sixteenth Order System

The F100 turbofan engine was chosen for initial analysis and [Ref. 9] describes this system in detail. The system "F", "G", "H" and "D" matrices were obtained from [Ref. 9: pp. 83-85] and the method of data entry to OPTSYSX is depicted in subroutine SETUP found in Appendix A. The selection of significant states is based upon a desired control bandwidth of 1 to 10 Hertz and following modal analysis these states were chosen as 1, 2, 5, 11 and 16. The following is a computer terminal system in which a fifth order model is generated.

BEGIN RECORDING OF TERMINAL SESSION

```
R; T=0.01/0.03 21:05:48
OPTRED
FILEDEF 05 TERM
FILEDEF 03 DISK REDUCI DATA A1
FILEDEF 06 DISK OPTRED LISTING A1
FILEDEF 07 DISK STATES DATA A1
FILEDEF 09 DISK OPTMAT DATA A1
FILEDEF 10 DISK OPTMATR DATA A1
GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLDP NONIMSL
LOAD OPTRED ( START
EXECUTION BEGINS...
```

```
OPTRED WILL COMPUTE A REDUCED ORDER MODEL FROM FULL SYSTEM
"F","G","H", AND "D" (IF INPUT) MATRICES.
```

```
THE FULL SYSTEM MATRICES MUST BE READ FROM A  FILE NAMED
"OPTMAT DATA" AS CREATED BY THE OPTSYSX PROGRAM.

YOU MUST ALSO ENTER THE DESIRED REDUCED ORDER (NUMBER OF STATES)
AND THE ACTUAL STATE #'S  (IN ASCENDING ORDER) WHICH REPRESENT
THE REDUCED MODEL. THE ORDER OF THE REDUCED MODEL MUST BE
LESS THAN THE ORDER OF THE FULL SYSTEM.

      DO YOU WISH TO CONTINUE?

           TYPE "YES" OR "NO".
 YES
      DO YOU WISH RESULTS TO SCREEN OR DISK?

     NOTE: A DATA FILE CONTAINING THE REDUCED ORDER DATA
           WILL BE GENERATED REGARDLESS OF YOUR ANSWER.

           TYPE "S" FOR SCREEN OR "D" FOR DISK.
 S
      THE ORDER OF THE FULL SYSTEM IS:  16
      THE NUMBER OF CONTROLS IS:  5
      THE NUMBER OF OBSERVATIONS IS:   7
      A "D" MATRIX WILL     BE INPUT.

      DO YOU WISH TO CONTINUE?

           TYPE "YES" OR "NO".
 YES
      ENTER THE DESIRED REDUCED ORDER OF THE "F" MATRIX.
 ?
 18
***** WARNING: REDUCED ORDER MUST BE GREATER THAN 0*****
                     AND LESS THAN  16

      ENTER THE DESIRED REDUCED ORDER OF THE "F" MATRIX.
 ?
 5
DO YOU WISH TO INPUT DESIRED STATES FOR YOUR REDUCED ORDER
MODEL FROM A DATA FILE?
```

```
  DATA FILE MUST BE NAMED "STATES DATA A1" IN FIXED FORMAT.
  THE READ FORMAT IS  "13I5" PER 72 CHARACTER LINE.
          TYPE "YES" OR "NO".
NO
ENTER THE "N" SIGNIFICANT STATES WHICH REPRESENT THE REDUCED
MODEL. ENTER STATE #'S IN ASCENDING ORDER.
      STATE #  1 =
?
1
      STATE #  2 =
?
2
      STATE #  3 =
?
3
      STATE #  4 =
?
11
      STATE #  5 =
?
16
    THE REDUCED MODEL STATES ARE:
     1    2    3   11   16
      DO YOU WISH TO CHANGE ANY OF THE SIGNIFICANT STATES?
          TYPE "YES" OR "NO".
YES
      ENTER THE N-TH STATE # TO BE CHANGED.
?
3
      ENTER NEW STATE #  3
5
```

```
   THE REDUCED MODEL STATES ARE:
    1     2     5    11    16

     DO YOU WISH TO CHANGE ANY OF THE SIGNIFICANT STATES?

          TYPE "YES" OR "NO".
NO

 (OUTPUT TO THE SCREEN FOLLOWS)

   THE DESIRED REDUCED ORDER IS:  5

   THE REDUCED MODEL STATES ARE:
    1     2     5    11    16

THE REDUCED PLANT MATRIX ("F") IS:

-3.0509D+00   2.4726D+00  -3.5898D+02   8.6913D+00   9.1832D-02
 1.2243D-01  -1.6271D+00   4.1060D+01   4.2898D+00   3.0229D-01
 2.7286D-03  -2.8331D-03  -7.4276D+00  -1.1695D-02  -9.4624D-03
 4.1314D+00  -5.3854D+00  -1.6257D+03  -5.3853D+01  -2.7911D+00
-1.0770D+00   2.2320D+00   1.1467D+03   1.9087D+01  -4.8897D+01

THE REDUCED CONTROL DISTRIBUTION MATRIX ("G") IS:

-3.5445D-02  -1.4080D+02  -9.3301D+01   2.3597D+01  -1.8379D+04
-4.4094D-01  -2.7221D+01   7.8306D+00  -1.0960D+01  -9.9711D+03
 1.7526D-02  -3.7646D+01   1.2113D-01  -7.8465D-02  -1.1931D+02
 1.9802D+01   3.1635D+02   4.9385D+01  -7.6813D+01   4.8330D+04
-1.8909D+00   7.2700D+01  -3.0436D+01   2.8867D+01   7.0035D+03

THE REDUCED OUTPUT DISTRIBUTION MATRIX ("H") IS:

 1.9281D-01   1.0750D-01   1.8753D+02   1.2088D+00   7.0796D-02
 6.4607D-03  -9.9442D-07  -9.1543D-03  -5.6722D-05  -1.9316D-05
 0.0          0.0          0.0          1.0000D+00   0.0
 1.2535D-05   1.3512D-05  -3.1266D-02  -1.3900D-04  -4.8476D-05
-4.8870D-05   1.4796D-04   1.0160D-03  -5.1681D-05   1.4491D-05
 1.6598D-05  -5.0355D-05  -3.7563D-02  -1.7308D-04  -5.9561D-05
 7.2415D-06  -8.1255D-07  -1.0493D-02  -4.9442D-05  -1.6687D-05
```

THE REDUCED FEEDFORWARD MATRIX ("D") IS:

-1.0377D-01	3.4001D+00	1.0378D+01	1.7019D+00	-4.1189D+03
-1.0929D-04	-2.2819D-01	3.1109D-01	4.8490D-03	-1.3105D+01
0.0	0.0	0.0	0.0	0.0
8.6209D-05	-8.5310D-03	-4.4940D-03	-9.5892D-05	-4.4513D-01
-1.5078D-05	4.1932D-03	-9.3797D-04	-3.9608D-03	9.6895D-01
1.0794D-04	5.6356D-03	-4.0947D-04	-7.9878D-04	-3.1434D-01
2.9257D-05	9.8556D-04	3.2771D-04	-9.4642D-05	-8.3787D-02

ANALYSIS COMPLETE...YOUR REDUCED SYSTEM DATA HAS BEEN SAVED
IN A FILE NAMED "OPTMATR DATA".

DO YOU DESIRE ANOTHER RUN?

IF YES, THE RESULTS OF THIS RUN WILL
REPLACE YOUR LAST OPTMATR DATA.

TYPE "YES" OR "NO".

NO

........OPTRED IS NOW TERMINATED........

R; T=0.38/0.77 21:06:56

RECORD OFF

END RECORDING OF TERMINAL SESSION

This is a typical computer terminal session in which the user requested output to the screen. Note also, that some of the user protection features of OPTRED are involked during this session.

Following this analysis, the OPTMATR data file was renamed OPTMAT data and OPTSYSX was executed using the reduced model data. The eigensystem was analyzed for desirable characteristics, including stability, and was compared to the full system eigenvalues. This analysis proved satisfactory and further comparison of the full and reduced

models was accomplished by obtaining Bode diagrams for the system open loop transfer functions. Figures 4.1 and 4.2 illustrate the full system and reduced system Bode plots respectively. The similarity, especially within the control bandwidth, is readily apparent. The frequency responses shown are for system input 1 and output 1. Analysis of other input/output combinations revealed the same favorable comparisons.

2. Example of a Ninety-eighth Order System

The X-29A control system includes a ninety-eighth order dynamics matrix. The measurement scaling matrix is constructed such that there are two measured outputs. These outputs represent the system with and without a notch filter, respectively. Full system matrix data as well as eigensystem data was obtained for this model from NASA-Edwards. Full system data was first entered into OPTSYSX using subroutine SETUP and the overall system was analyzed. Sixty significant states were identified and are shown in the following recorded terminal session. This example also illustrates program execution where no "D" matrix is input and the reduced system states are entered by means of file data. During this session the user also chooses to have the general reduced model data printed to a listing file.

```
        BEGIN RECORDING OF TERMINAL SESSION
R; T=0.01/0.02 12:23:56
OPTRED
FILEDEF 05 TERM
FILEDEF 06 DISK OPTRED LISTING A1
FILEDEF 07 DISK STATES DATA A1
FILEDEF 09 DISK OPTMAT DATA A1
FILEDEF 10 DISK OPTMATR DATA A1
GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLDP NONIMSL
```

```
LOAD OPTRED ( START
EXECUTION BEGINS...

OPTRED WILL COMPUTE A REDUCED ORDER MODEL FROM FULL SYSTEM
"F","G","H", AND "D" (IF INPUT) MATRICES.

THE FULL SYSTEM MATRICES MUST BE READ FROM A  FILE NAMED
"OPTMAT DATA" AS CREATED BY THE OPTSYSX PROGRAM.

YOU MUST ALSO ENTER THE DESIRED REDUCED ORDER (NUMBER OF STATES)
AND THE ACTUAL STATE #'S  (IN ASCENDING ORDER) WHICH REPRESENT
THE REDUCED MODEL. THE ORDER OF THE REDUCED MODEL MUST BE
LESS THAN THE ORDER OF THE FULL SYSTEM.

     DO YOU WISH TO CONTINUE?

          TYPE "YES" OR "NO".
YES
     DO YOU WISH RESULTS TO SCREEN OR DISK?

    NOTE: A DATA FILE CONTAINING THE REDUCED ORDER DATA
          WILL BE GENERATED REGARDLESS OF YOUR ANSWER.

          TYPE "S" FOR SCREEN OR "D" FOR DISK.
D
     THE ORDER OF THE FULL SYSTEM IS:  98
     THE NUMBER OF CONTROLS IS:  1
     THE NUMBER OF OBSERVATIONS IS:   2
     A "D" MATRIX WILL NOT BE INPUT.

     DO YOU WISH TO CONTINUE?

          TYPE "YES" OR "NO".
YES
     ENTER THE DESIRED REDUCED ORDER OF THE "F" MATRIX.
?
60
DO YOU WISH TO INPUT DESIRED STATES FOR YOUR REDUCED ORDER
MODEL FROM A DATA FILE?
```

```
    DATA FILE MUST BE NAMED "STATES DATA A1" IN FIXED FORMAT.
    THE READ FORMAT IS  "13I5" PER 72 CHARACTER LINE.
         TYPE "YES" OR "NO".
YES

  THE REDUCED MODEL STATES ARE:
 1    2    3    4    5    6   12   13   14   15   43   44   45
46   47   48   49   50   51   52   53   54   55   62   63   64
65   66   67   68   69   70   71   72   73   74   75   76   77
78   79   80   81   82   83   84   85   86   87   88   89   90
91   92   93   94   95   96   97   98

    DO YOU WISH TO CHANGE ANY OF THE SIGNIFICANT STATES?

         TYPE "YES" OR "NO".
NO

ANALYSIS COMPLETE...YOUR REDUCED SYSTEM DATA HAS BEEN SAVED
            IN A FILE NAMED "OPTMATR DATA".

              DO YOU DESIRE ANOTHER RUN?

         IF YES, THE RESULTS OF THIS RUN WILL
            REPLACE YOUR LAST OPTMATR DATA.

                   TYPE "YES" OR "NO".
NO

        ........OPTRED IS NOW TERMINATED........

R; T=17.70/18.51 12:25:07
RECORD OFF
END RECORDING OF TERMINAL SESSION
```

The eigensystems for both systems were analyzed for stability and other desireable characteristics. Bode plots were obtained for both input/output combinations and these

graphs for the full and reduced systems are included as Figures 4.3 through 4.10. The input/output combination (1/1) shows the response of the system without the notch filter while the (1/2) input/output combination depicts the frequency response of the system after the addition of the notch filter. These figures clearly indicate excellent agreement between the full and reduced order systems in the frequency domain.

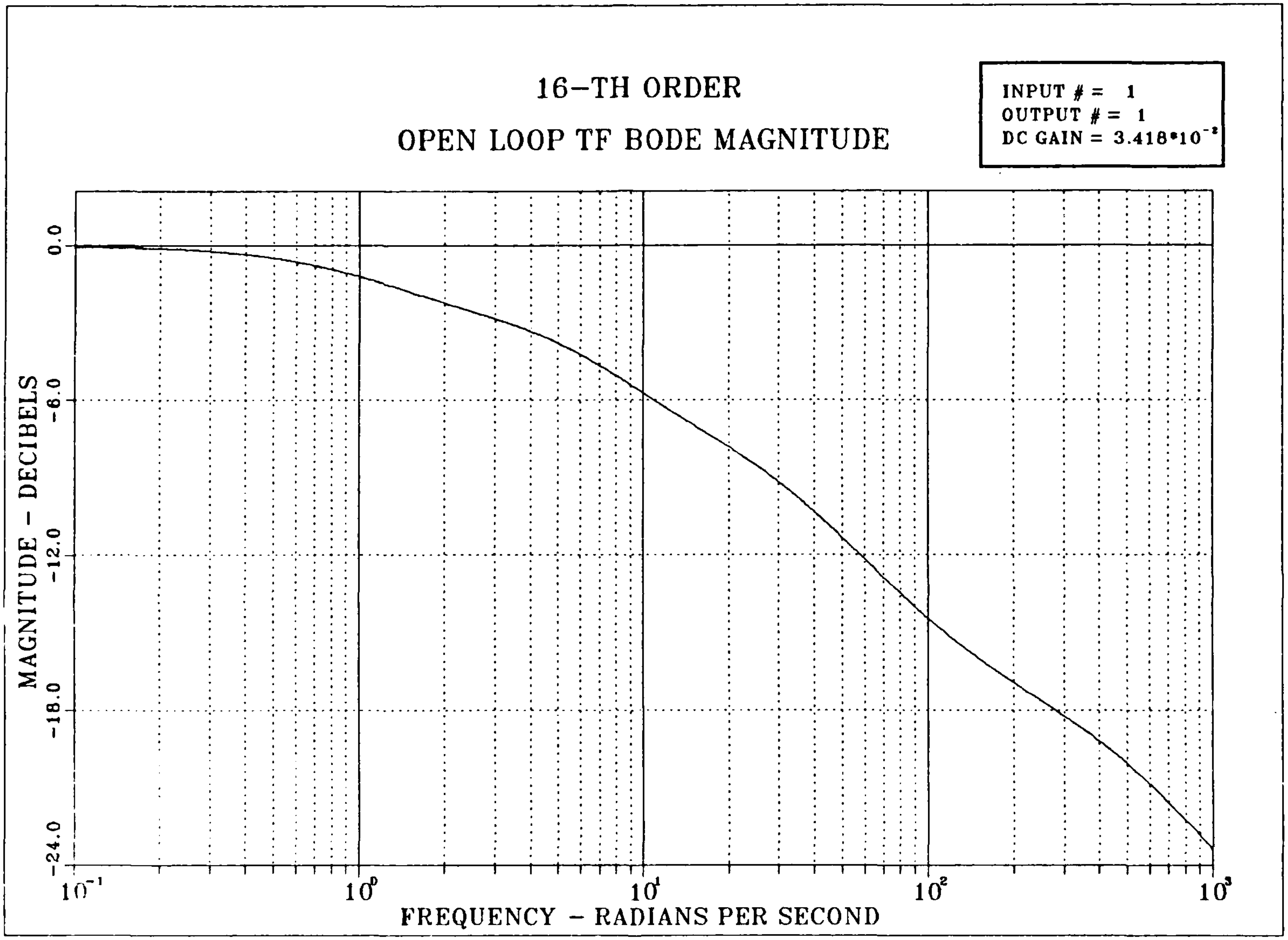


Figure 4.1 Full System Bode Plot for the F100 Engine.

```
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
      GO TO 70
60    WRITE (5,880)
      GO TO 50
70    CONTINUE
      IF (IANS.EQ.IZ) GO TO 560
C-----------------------------------------------------ISET-------------------
80    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,910)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 90
      GO TO 100
90    WRITE (5,880)
      GO TO 80
100   CONTINUE
      IF (IANS.EQ.IY) ISET=1
C-----------------------------------------------------------------------------
C  INITIALIZE SYSTEM FLAGS.                                                  =
C-----------------------------------------------------------------------------
10    CONTINUE
      IRET=0
      IOL=0
      IQ=0
      IR=0
      ISS=0
      IM=0
      ITF1=0
      ITF2=0
      ITF3=0
      IFDFW=0
      IE=0
      IDSTAB=0
      IDEBUG=0
      IPSD=0
      IYU=0
      INORM=0
      IREG=0
      NS=0
      NC=0
      NOB=0
      NG=0
      IRDMAT=0
C-----------------------------------------------------IOL--------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,570)
      CALL RDINT (IANS)
      IOL=IANS-1
      IF (IOL.EQ.2) GO TO 350
```

```
C     5,D(32,32),DSTORE(32,32),JCF(64),RES(64),AY(32,32),BB(64),CC(64),CP
C     6(32),GW(64,64),GV(64,64),HY(64,64),HU(64,64),PRTT(16,16)
C-----------------------------------------------------------------------
      EQUIVALENCE (W11(1,1),GW(1,1)), (W11(1,1),GV(1,1)), (W21(1,1),HY(1
     1,1)), (W21(1,1),HU(1,1))
C-----------------------------------------------------------------------
      COMMON /PROG/ IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,IDEB
     1UG,ISET,IREG,IPSD,IYU,INORM
C-----------------------------------------------------------------------
      DATA IY/'Y'/,IZ/'N'/
C-----------------------------------------------------------------------
C  SUPPRESS INDIVIDUAL UNDERFLOW, OVERFLOW, DIVIDE CHECK, AND DECIMAL  =
C  CONVERT ERROR MESSAGES; PROVIDE SUMMARY OF ERRORS ONLY.             =
C-----------------------------------------------------------------------
      CALL ERRSET (207,256,-1,1,1,209)
      CALL ERRSET (215,256,-1,1)
C-----------------------------------------------------------------------
C  INITIALIZE SAVE FLAGS.                                              =
C-----------------------------------------------------------------------
      ISAF=0
      ISAG=0
      ISAH=0
      IGAM=0
      ISAD=0
      ISAA=0
      ISAB=0
      ISET=0
C--------------------------------------------SCRNA----------------------
5     CALL FRTCMS ('CLRSCRN ')
      WRITE (5,885)
      CALL RDINT (IANS)
      IF(IANS.EQ.1) GO TO 20
      IF(IANS.EQ.2) GO TO 10
      GO TO 5
C-----------------------------------------------SCRN1-------------------
20    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,890)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 30
      GO TO 40
30    WRITE (5,880)
      GO TO 20
40    CONTINUE
      IF (IANS.EQ.IZ) GO TO 560
C-----------------------------------------------SCRN2-------------------
50    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,900)
      CALL RDCHAR (IANS)
```

APPENDIX A

THE OPTSYSX PROGRAM AS MODIFIED BY THIS THESIS

```
C*******************************************************************
C*******************************************************************
C************                                           ************
C**********                 OPTSYSX                      **********
C********              BY JOHN G. HODEN                    ********
C********                                                  ********
C********    THIS PROGRAM IS A COMPLETELY INTERACTIVE      ********
C********    OPTIMAL SYSTEMS CONTROL DESIGN/SYNTHESIS      ********
C********    PROGRAM CAPABLE OF HANDLING VERY LARGE (80X80)+ ********
C********    MULTIVARIABLE SYSTEMS OF LINEAR EQUATIONS.    ********
C********                                                  ********
C**********          VERSION 1.8    11 MAR 1984          **********
C************           MODIFIED BY AL DIEL             ************
C************          MODIFIED BY M.H. LAPTAS          ************
C************          MODIFIED BY S.W. NELSON          ************
C*******************************************************************
C*******************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
C-------------------------------------------------------------------
      INTEGER IANS,IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDEBUG,ISET,
     1IPSD,IYU,INORM,NS,NC,NOB,NG,IREG,IDSTAB,IRET,NROW,NCOL,ISAF,ISAH,I
     2SAG,ISAD,IGAM,IRDMAT,ZERO,ONE,ITFX,NR
C-------------------------------------------------------------------
C     INCREASED DIMENSIONS -- LARGE ORDER SYSTEM.
C-------------------------------------------------------------------
      DIMENSION ACL(98,98),B(13,13),BA(98,98),CI(98),CR(98),CQ(98,98),
     *CWI(98),CWR(98),FBGC(13,98),FBGE(98,13),G(98,98),GM(98,98),
     *PRO(98,98),RC(13,13),SC(98,98),WR(196),WI(196),W11(98,98),
     *W21(98,98),X(196,196),GN(98,98),HO(13,98),D1(196),D2(196),
     *RM(196,196),Q(1,1),GAM(98,1),WNORM(98,98),WNORMI(98,98),
     *DESTAB(98),AA(98,98),BM(98,13),CM(13,98),D(98,13),DSTORE(98,98),
     *JCF(196),RES(196),AY(13,13),BB(196),CC(196),CP(98),GW(196,1),
     *GV(196,13),HY(13,196),HU(13,196),PRTT(16,16)
C-------------------------------------------------------------------
C     STANDARD PROGRAM DIMENSIONS.
C-------------------------------------------------------------------
C     DIMENSION ACL(32,32),B(32,32),BA(32,32),CI(32),CR(32),CQ(32,32),CW
C    1I(32),CWR(32),FBGC(32,32),FBGE(32,32),G(32,32),GM(32,32),PRO(32,32
C    2),RC(32,32),SC(32,32),WR(64),WI(64),W11(32,32),W21(32,32),X(64,64)
C    3,GN(32,32),HO(32,32),D1(64),D2(64),RM(64,64),Q(32,32),GAM(32,32),W
C    4NORM(32,32),WNORMI(32,32),DESTAB(32),AA(32,32),BM(32,32),CM(32,32)
```

thesis. The addition of this program to the analysis technique will provide a further means of verification of the accuracy and fidelity of the reduced order model.

2. Interface with Other Analysis Programs

The Naval Postgraduate School's Aeronautical Engineering Department has developed a collection of analysis programs including OPTSYSX, OPTGRAPH and OPTCALC. These programs exist in a single "module" which enables the user to interactively execute these programs during a single computer terminal session. The feasibility of adding OPTRED to this module should be investigated. If necessary, these programs would require re-dimensioning and could create a very large memory size requirement. Additions and modifications would also be necessary in the OPTSYSX EXEC to enable the execution of OPTRED and then transfer execution to OPTSYSX for the analysis of the reduced model. This controlling EXEC must also manage the data files upon which the individual programs depend.

3. Alternate Order Reduction Techniques

Finally, other methods of order reduction should be investigated. Comparisons of reduced order models which are generated by different techniques would provide the controls designer with additional means of reduced model verification and analysis.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The computational capabilities of OPTRED were tested by analyzing two significant large order systems. The determination of the validity of both full order and reduced order results was greatly simplified by the eigensystem data for the X-29A provided by NASA-Edwards. As discussed by Laptas [Ref. 6], OPTSYSX failed to produce valid open-loop transfer function zeroes when analyzing very large systems. However, through modal analysis and with the aid of the NASA data, the extraneous zeroes were easily identified and the open loop data created by OPTSYSX was corrected.

The results of OPTSYSX and OPTGRAPH indicate a very strong similarity between the full system and the reduced system generated by OPTRED both in stability and in the frequency domain. The combined analytic capabilities of OPTSYSX, OPTGRAPH and OPTRED provide a powerful tool for the controls designer for use in the analysis, development and the implementation of complex large order systems.

B. RECOMMENDATIONS

Based upon the results of this thesis the following areas are recommended for further study and research.

1. Comparison of System Time Response

Although the reduced model must closely resemble the full system in the frequency domain, it should also compare favorably in the time domain. OPTCALC, the time response program, requires several modifications to enable its use for analysis of systems as large as those presented in this

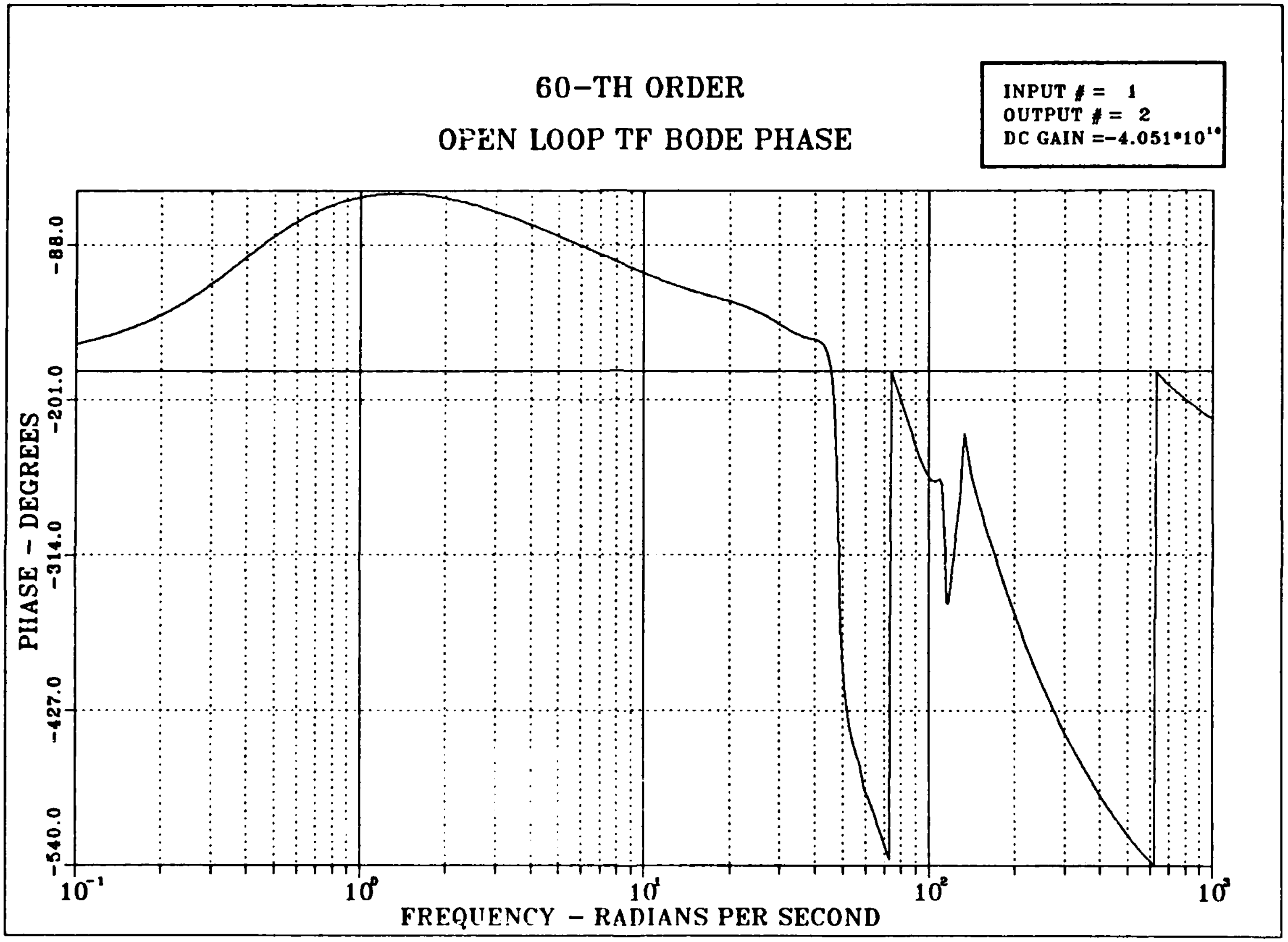


Figure 4.10 X-29A Reduced System With Notch Filter.

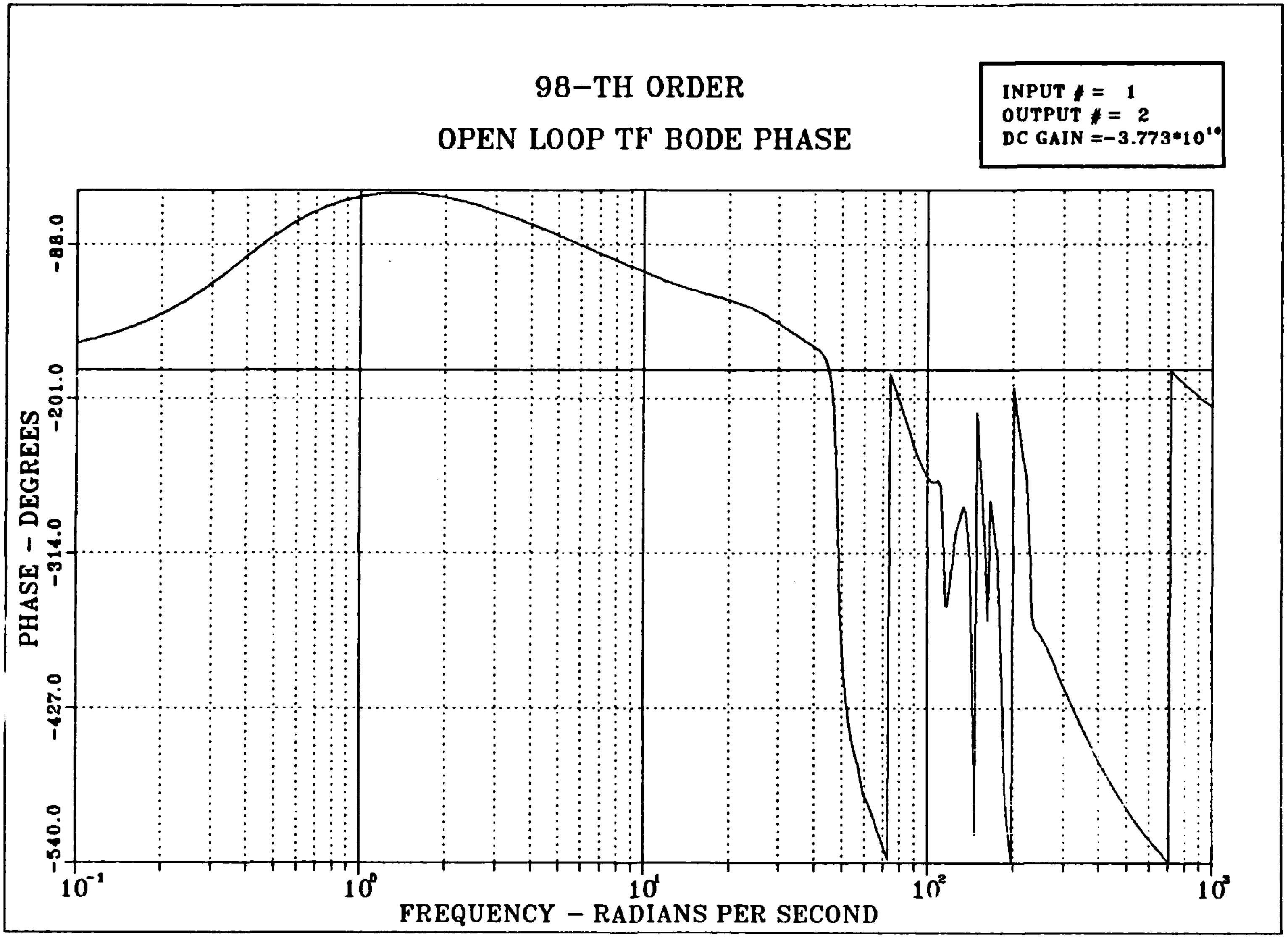


Figure 4.9 X-29A Full System With Notch Filter.

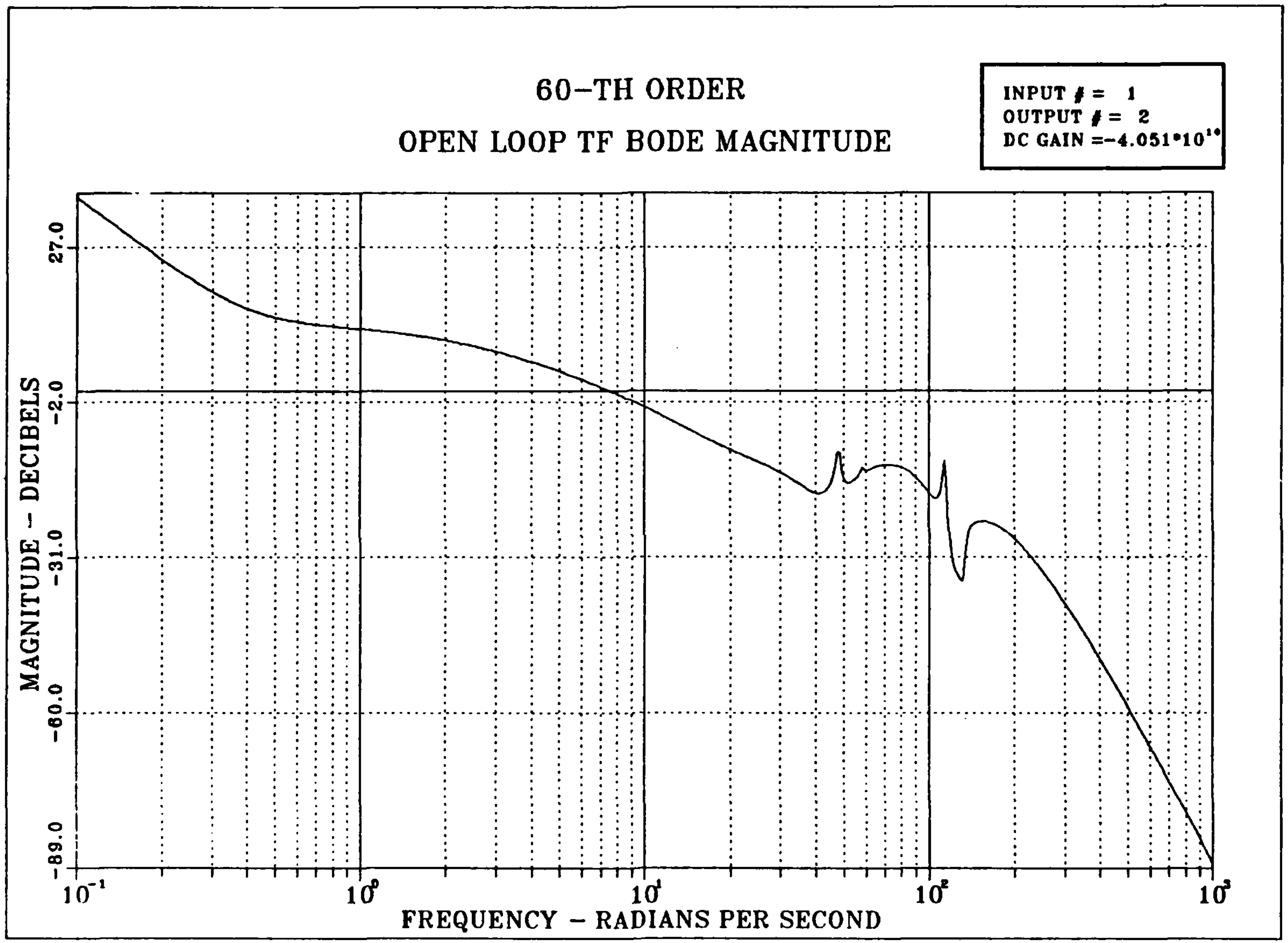


Figure 4.8   X-29A Reduced System With Notch Filter.

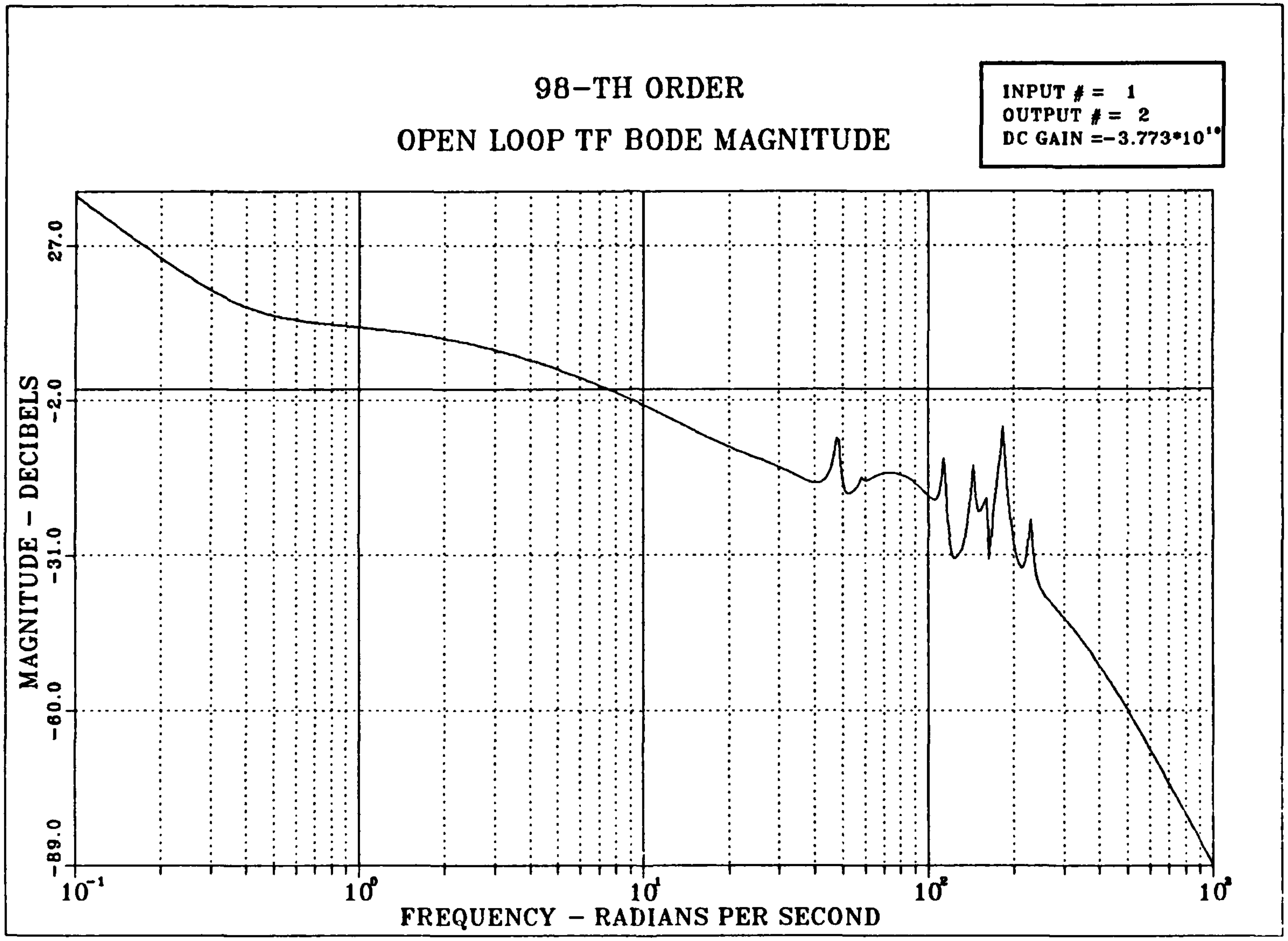


Figure 4.7 X-29A Full System With Notch Filter.

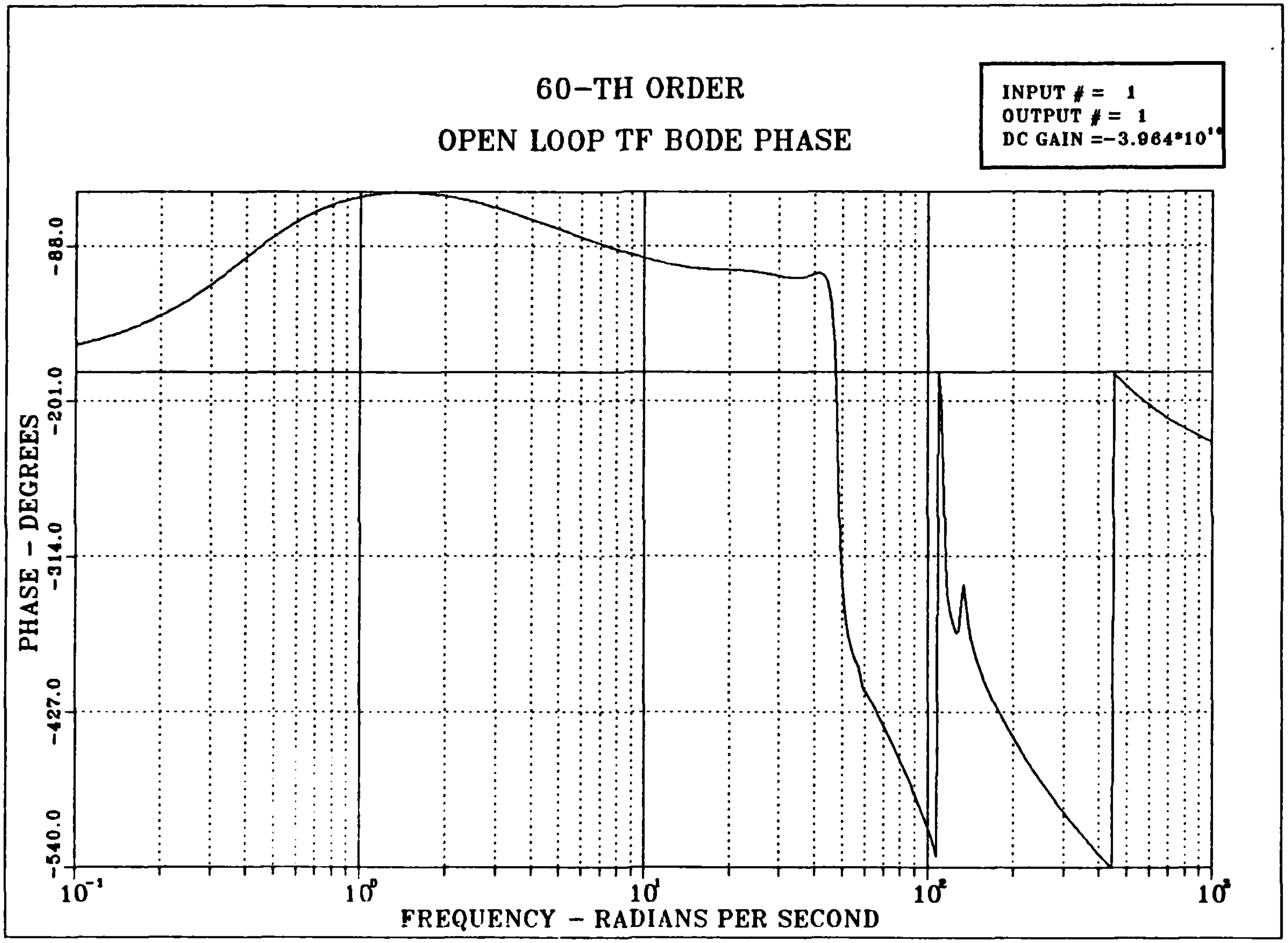


Figure 4.6 X-29A Reduced System Without Notch Filter.

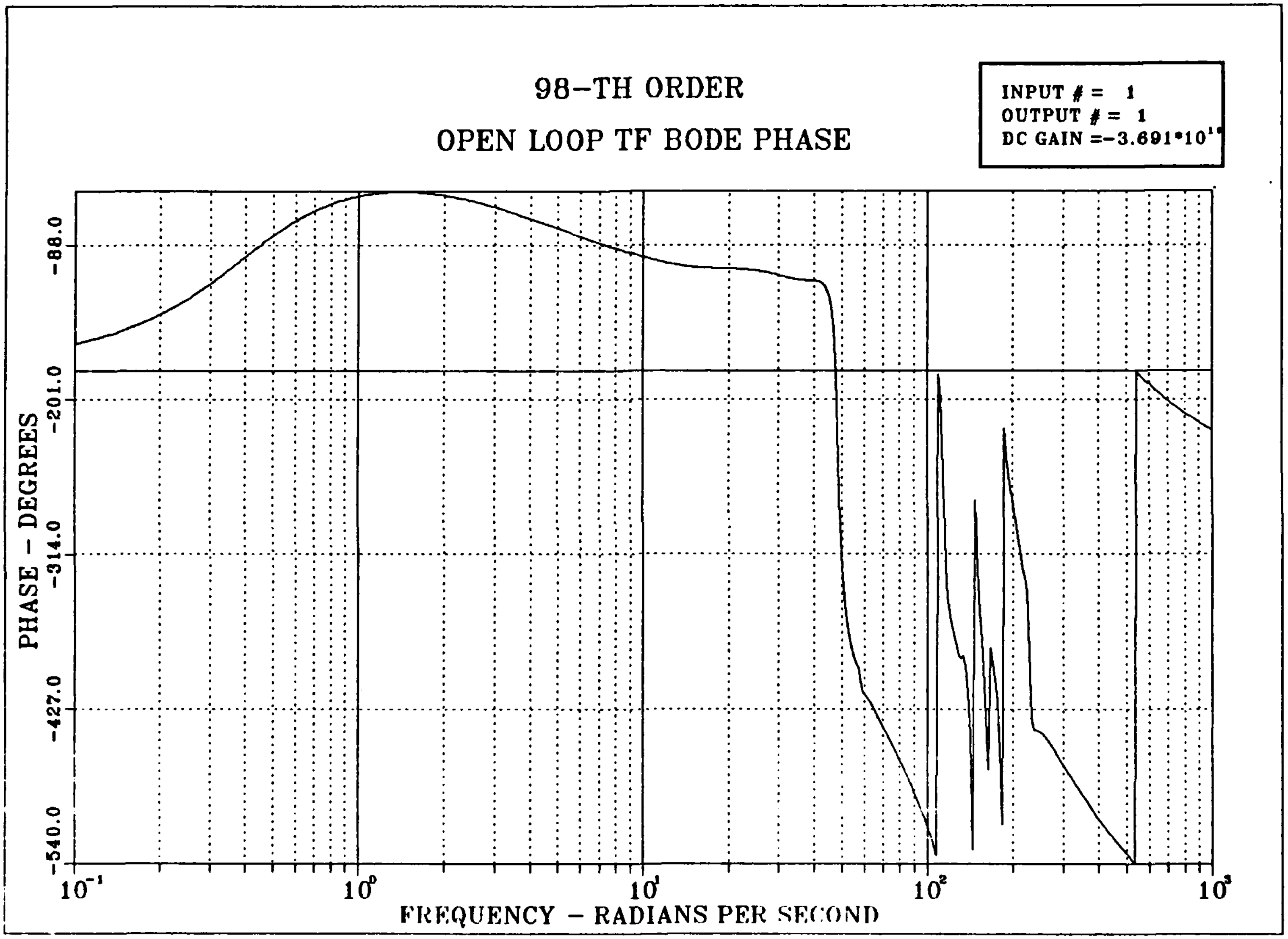


Figure 4.5 X-29A Full System Without Notch Filter.

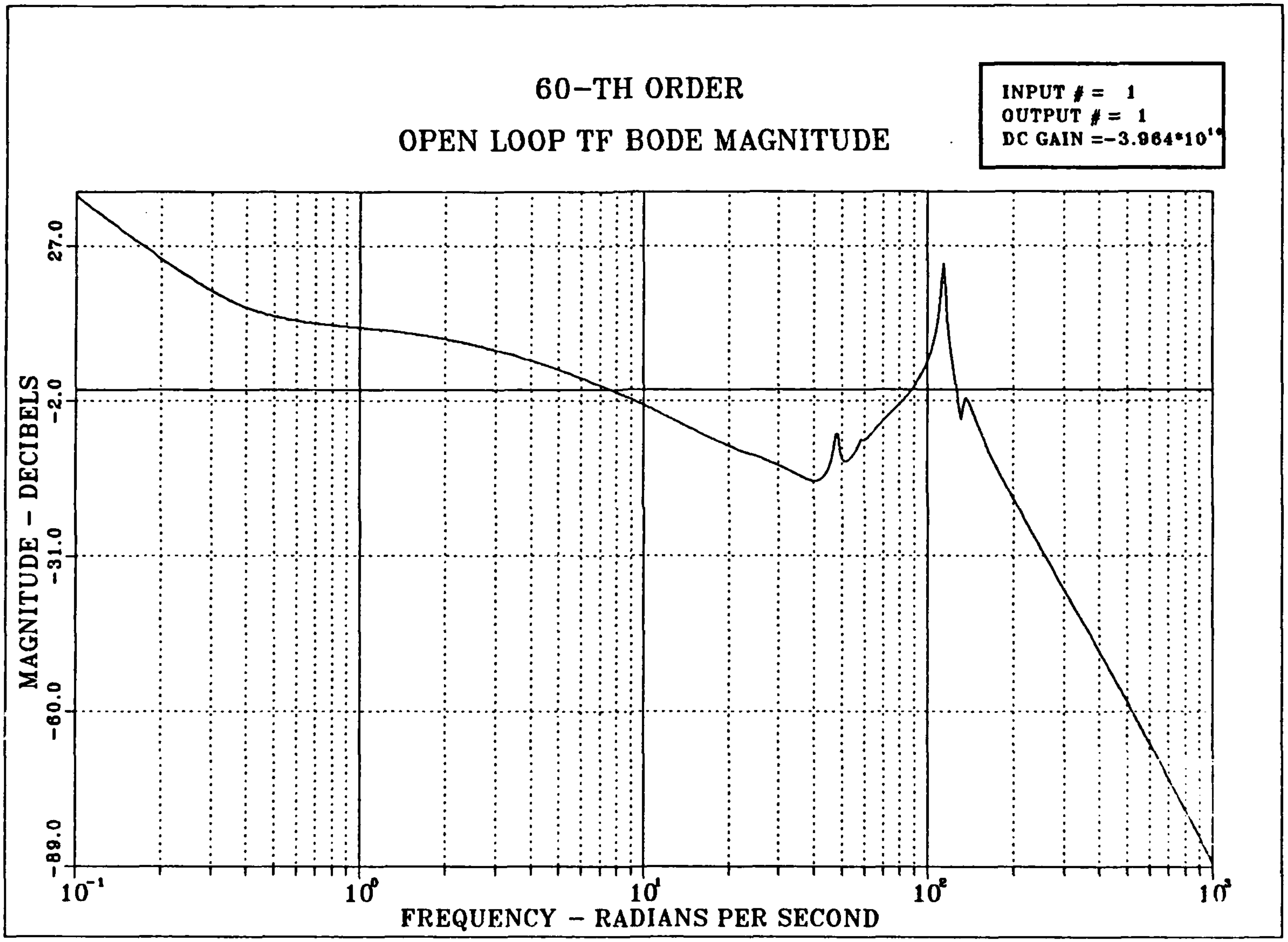


Figure 4.4 X-29A Reduced System Without Notch Filter.

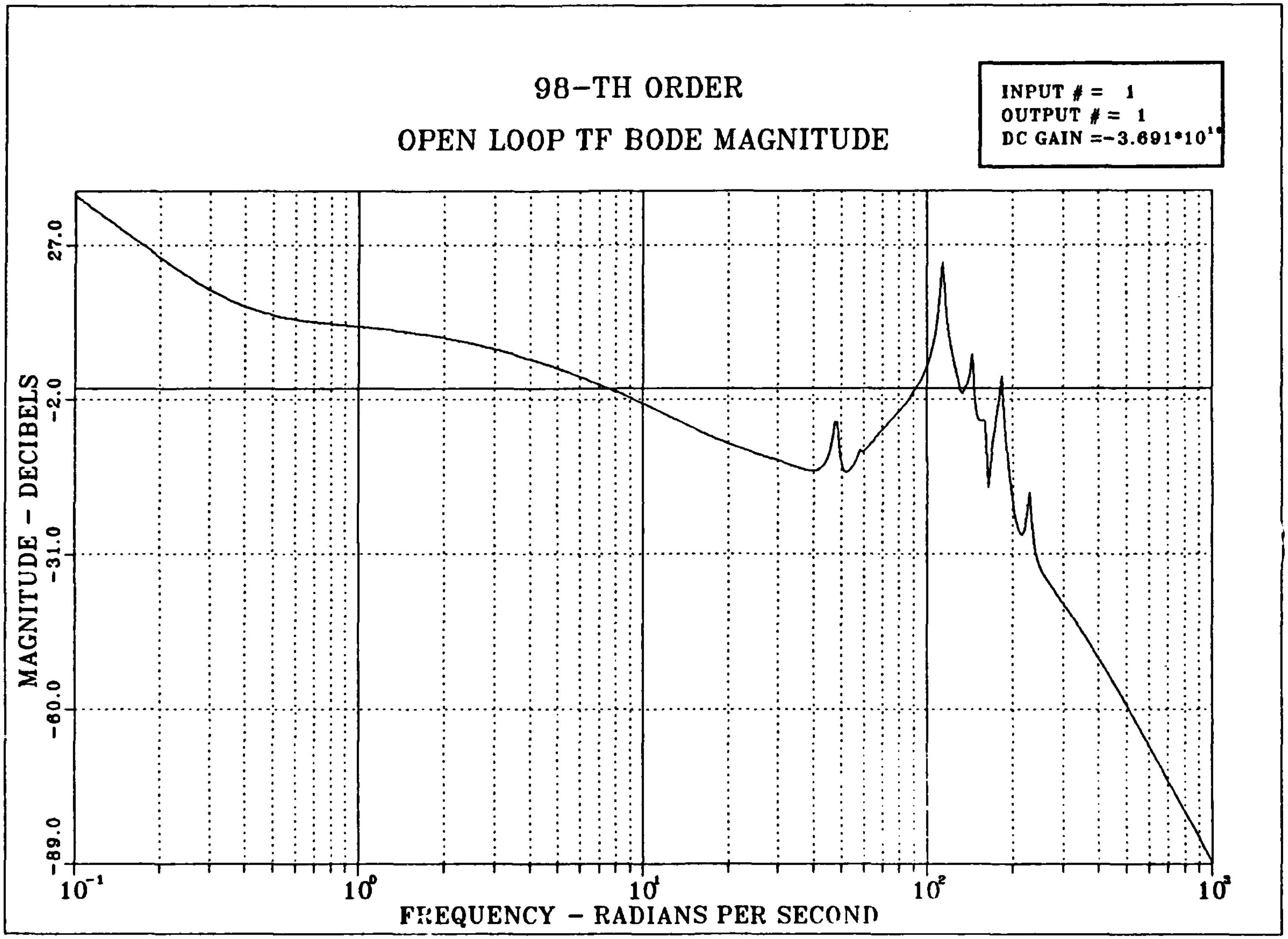


Figure 4.3 X-29A Full System Without Notch Filter.

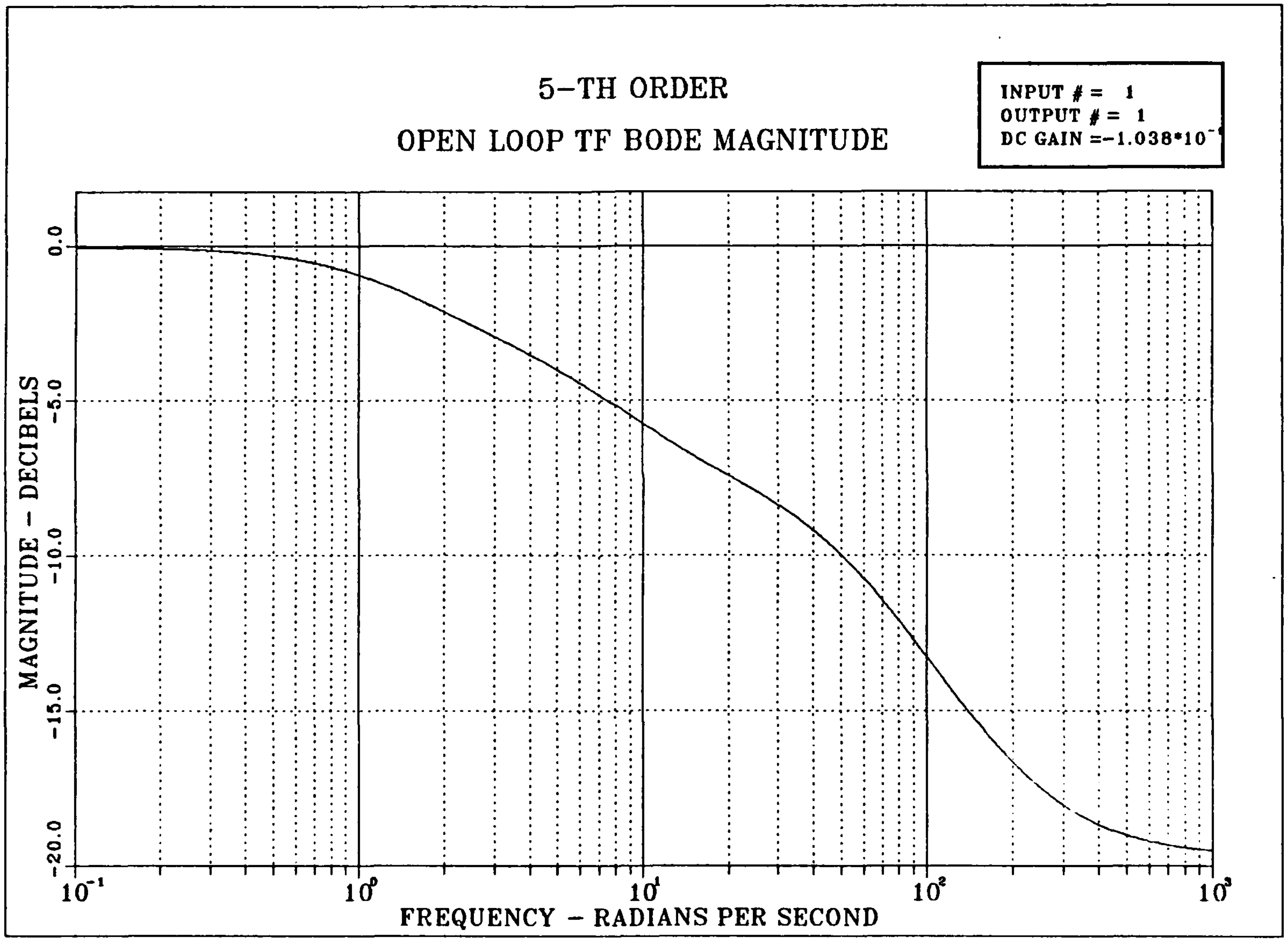


Figure 4.2 Reduced System Bode Plot for the F100 Engine.

```
      IF (IOL.EQ.3) GO TO 200
C--------------------------------------------------IQ--------------------
      CALL FRTCMS ('CLRSCRN ')
110   WRITE (5,580)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 120
      GO TO 130
120   WRITE (5,880)
      GO TO 110
130   CONTINUE
      IF (IANS.EQ.IY) IQ=1
      IF (IANS.EQ.IZ) IQ=0
C--------------------------------------------------IR--------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,590)
      CALL RDINT (IANS)
      IR=IANS-1
C--------------------------------------------------ISS-------------------
      CALL FRTCMS ('CLRSCRN ')
140   WRITE (5,600)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 150
      GO TO 160
150   WRITE (5,880)
      GO TO 140
160   CONTINUE
      IF (IANS.EQ.IY) ISS=1
      IF (IANS.EQ.IZ) ISS=0
C--------------------------------------------------IM--------------------
170   WRITE (5,610)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 180
      GO TO 190
180   WRITE (5,880)
      GO TO 170
190   CONTINUE
      IF (IANS.EQ.IY) IM=1
      IF (IANS.EQ.IZ) IM=0
200   CONTINUE
      IF (IOL.EQ.3) IM=1
C--------------------------------------------------IFDFW-----------------
      CALL FRTCMS ('CLRSCRN ')
210   WRITE (5,650)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 220
      GO TO 230
220   WRITE (5,880)
      GO TO 210
```

```
230   CONTINUE
      IF (IANS.EQ.IY) IFDFW=1
      IF (IANS.EQ.IZ) IFDFW=0
      IF (IOL.EQ.3) GO TO 350
C-------------------------------------------------IDSTAB----------------
      CALL FRTCMS ('CLRSCRN ')
240   WRITE (5,670)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 250
      GO TO 260
250   WRITE (5,880)
      GO TO 240
260   CONTINUE
      IF (IANS.EQ.IY) IDSTAB=1
      IF (IANS.EQ.IZ) IDSTAB=0
C-------------------------------------------------IDEBUG----------------
270   WRITE (5,680)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 280
      GO TO 290
280   WRITE (5,880)
      GO TO 270
290   CONTINUE
      IF (IANS.EQ.IY) IDEBUG=1
      IF (IANS.EQ.IZ) IDEBUG=0
300   CONTINUE
C-------------------------------------------------IREG------------------
      CALL FRTCMS ('CLRSCRN ')
320   WRITE (5,710)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 330
      GO TO 340
330   WRITE (5,880)
      GO TO 320
340   CONTINUE
      IF (IANS.EQ.IY) IREG=1
      IF (IANS.EQ.IZ) IREG=0
350   CALL RDMATF(NS,NC,NOB,NG,ISAF,ISAG,ISAH,ISAD,IGAM,ISAA,ISAB,IRDMAT
     1,IFDFW)
      IF ((ISAF.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 352
C-------------------------------------------------NS--------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,720)
      CALL RDREAL (ANSR)
      NS=IDINT(ANSR)
352   IF (IOL.EQ.2) GO TO 360
      IF ((ISAG.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 354
C-------------------------------------------------NC--------------------
```

```
      WRITE (5,730)
      CALL RDREAL (ANSR)
      NC=IDINT(ANSR)
354   IF ((ISAH.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 356
C------------------------------------------------NOB--------------------
      WRITE (5,740)
      CALL RDREAL (ANSR)
      NOB=IDINT(ANSR)
356   IF ((IGAM.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 360
C------------------------------------------------NG---------------------
      WRITE (5,750)
      CALL RDREAL (ANSR)
      NG=IDINT(ANSR)
360   CONTINUE
      IF (IOL.EQ.2) GO TO 364
      IF (IOL.EQ.3) GO TO 310
C------------------------------------------------IPSD-------------------
      IF(IREG.NE.0.OR.NC.EQ.0) GO TO 310
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,690)
      CALL RDINT (IANS)
      IPSD=IANS
      IF (IPSD.EQ.3) IPSD=0
      IF (IPSD.EQ.0) GO TO 310
C------------------------------------------------IYU--------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,700)
      CALL RDINT (IANS)
      IYU=IANS-1
C------------------------------------------------INORM------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,820)
      CALL RDREAL (ANSR)
      INORM=IDINT(ANSR)
310   CONTINUE
C------------------------------------------------ITF1-------------------
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,62C)
      CALL RDINT (IANS)
      ITF1=IANS-1
      IF (IOL.EQ.3) GO TO 362
C------------------------------------------------ITF2-------------------
      IF(IREG.NE.1) GO TO 315
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,630)
      CALL RDINT (IANS)
      ITF2=IANS-1
C------------------------------------------------ITF3-------------------
```

```
315   IF (IREG.NE.0 .OR. (NC.EQ.0 .OR. NG.EQ.0)) GO TO 362
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,640)
      CALL RDINT (IANS)
      ITF3=IANS-1
C-----------------------------------------------------IE--------------------
362   CONTINUE
      IF((ITF1+ITF2+ITF3).EQ.0) GO TO 364
      CALL FRTCMS ('CLRSCRN ')
      WRITE (5,660)
      CALL RDREAL (ANSR)
      IE=IDINT(ANSR)
C-----------------------------------------------------FLAG SETTINGS----------
364   CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      WRITE (6,760)
      WRITE (6,770)
      WRITE (6,780) IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDEBUG,ISET
     1,IDSTAB
      WRITE (6,790)
      WRITE (6,800) IPSD,IYU,INORM,IREG,NS,NC,NOB,NG
      WRITE (6,810) NS,NC,NCB,NG
C-----------------------------------------------------OPTGRAPH DATA FILES ---
      ZERO = 0
      ONE = 1
      REWIND 10
      ITFX = 1
      WRITE (10,1000) ZERO,ONE,NS,NC,NOB,IE,ITFX
      REWIND 1
      ITFX = 2
      WRITE (1,1000) ZERO,ONE,NS,NG,NOB,IE,ITFX
      REWIND 4
      ITFX = 3
      WRITE (4,1000) ZERO,ONE,NS,NOB,NC,IE,ITFX
C-----------------------------------------------------BEGIN CALCULATIONS-----
      N2=2*NS
      CALL INNER (NS,NC,NOB,NG,N2,ACL,B,BA,CI,CR,CQ,CWI,CWR,D,FBGC,FBGE,
     1G,GAM,GM,GN,HO,D1,D2,ERO,RM,RC,Q,SC,WR,WI,W11,W21,X,WNORM,WNORMI,D
     2ESTAB,AA,BM,CM,JCP,RES,AY,BB,CC,CP,GW,GV,HY,HU,DSTORE,ISAF,ISAH,IS
     3AG,ISAD,IGAM,IRET,PRTT,NROW,NCOL,IRDMAT,ISAA,ISAB)
C------------------------------------------------------OPTGROL DATA----------
      IF((ITF1.EQ.1).OR.(ITF1.EQ.2)) GO TO 396
      END FILE 10
      REWIND 10
      ITFX = 1
      WRITE (10,1000) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ITFX
C------------------------------------------------------OPTGRNO DATA----------
396   IF((ITF2.EQ.1).OR.(ITF2.EQ.2)) GO TO 397
```

```
      END FILE 1
      REWIND 1
      ITFX = 2
      WRITE (1,1000) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ITFX
C----------------------------------------------OPTGRCM DATA-----------
397   IF((ITF1.EQ.3).OR.(ITF3.EQ.2)) GO TO 398
      END FILE 4
      REWIND 4
      ITFX = 3
      WRITE (4,1000) ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ITFX
C---------------------------------------------------------------------
398   IF(IRET.EQ.1) GO TO 370
      CALL WRTMAT(BA,G,HO,D,GAM,FBGC,FBGE,AY,B,NS,NC,NOB,NG,IFDFW)
C----------------------------------------------IRET-------------------
370   WRITE (5,830)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 380
      GO TO 390
380   WRITE (5,880)
      GO TO 370
390   CONTINUE
C----------------------------------------------RESET OPGRAPH DATA FILE
      IF(IANS.EQ.IZ) GO TO 395
      END FILE 10
      REWIND 10
      END FILE 1
      REWIND 1
      END FILE 4
      REWIND 4
C---------------------------------------------------------------------
395   IF (IANS.EQ.IY) GO TO 400
      IF (IANS.EQ.IZ) GO TO 560
C----------------------------------------------ISAF-------------------
400   CONTINUE
      IF (IRET.EQ.1) GO TO 10
      IF (ISET.EQ.1) GO TO 10
      CALL FRTCMS ('CLRSCRN ')
410   WRITE (5,840)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 420
      GO TO 430
420   WRITE (5,880)
      GO TO 410
430   CONTINUE
      IF (IANS.EQ.IY) ISAF=1
      IF (IANS.EQ.IZ) ISAF=0
C----------------------------------------------ISAH-------------------
      IF (NOB.EQ.0) GO TO 470
```

```
      CALL FRTCMS ('CLRSCRN ')
440   WRITE (5,850)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 450
      GO TO 460
450   WRITE (5,880)
      GO TO 440
460   CONTINUE
      IF (IANS.EQ.IY) ISAH=1
      IF (IANS.EQ.IZ) ISAH=0
470   CONTINUE
C-----------------------------------------------------ISAG------------------
      IF (NC.EQ.0) GO TO 510
      CALL FRTCMS ('CLRSCRN ')
480   WRITE (5,860)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 490
      GO TO 500
490   WRITE (5,880)
      GO TO 480
500   CONTINUE
      IF (IANS.EQ.IY) ISAG=1
      IF (IANS.EQ.IZ) ISAG=0
510   CONTINUE
C-----------------------------------------------------ISAD------------------
      IF (IFDFW.EQ.0) GO TO 515
      CALL FRTCMS ('CLRSCRN ')
511   WRITE (5,865)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 513
      GO TO 514
513   WRITE (5,880)
      GO TO 511
514   CONTINUE
      IF (IANS.EQ.IY) ISAD=1
      IF (IANS.EQ.IZ) ISAD=0
515   CONTINUE
C-----------------------------------------------------IGAM------------------
      IF (NG.EQ.0) GO TO 550
      CALL FRTCMS ('CLRSCRN ')
520   WRITE (5,870)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 530
      GO TO 540
530   WRITE (5,880)
      GO TO 520
540   CONTINUE
      IF (IANS.EQ.IY) IGAM=1
```

```
      IF (IANS.EQ.IZ) IGAM=0
550   CONTINUE
C------------------------------------------------------ISAA--------------------
      CALL FRTCMS ('CLRSCRN ')
551   WRITE (5,872)
      CALL RDCHAR (IANS)
      IF ((IANS.EQ.IY).OR.(IANS.EQ.IZ)) GO TO 553
      WRITE (5,880)
      GO TO 551
553   CONTINUE
      IF (IANS.EQ.IY) ISAA=1
      IF (IANS.EQ.IZ) ISAA=0
C------------------------------------------------------ISAB--------------------
      CALL FRTCMS ('CLRSCRN ')
555   WRITE (5,874)
      CALL RDCHAR (IANS)
      IF ((IANS.EQ.IY).OR.(IANS.EQ.IZ)) GO TO 557
      WRITE (5,880)
      GO TO 555
557   CONTINUE
      IF (IANS.EQ.IY) ISAB=1
      IF (IANS.EQ.IZ) ISAB=0
      GO TO 10
C------------------------------------------------------TERMINATE---------------
560   WRITE (5,920)
      STOP
C------------------------------------------------------------------------------
885   FORMAT (5X,33HIN CASE OF PROBLEMS CALL EXT 2826,///,10X,42H1  DEFI
     1NES MATRIX EQUATIONS USED IN OPTSYS,/,15X,32HAND ALTERNATE DATA IN
     2PUT METHODS,//,10X,26H2  START INTERACTIVE INPUT,///,15X,13HENTER
     3 1 OR 2,///,5X,59HNORMALLY TWO SUCCESSIVE NULL ENTERS TERMINATES T
     4HE PROGRAM.)
890   FORMAT (5X,59HOPTSYSX IS A COMPLETELY INTERACTIVE OPTIMAL SYSTEMS
     1CONTROL,/,8X,55HPROGRAM. IT WILL SOLVE NUMEROUS CONTROL PROBLEMS O
     2N THE,/,8X,45HFOLLOWING TYPES OF SYSTEMS CONTROL EQUATIONS:,//,15X
     3,35HXDOT = {F}*X + {G}*U + {GAM}*(W+W0),//,20X,22HMEASUREMENT EQUA
     4TION--,//,15X,21HZ = {H}*X + {D}*U + V,//,20X,29HREGULATOR PERFORM
     5ANCE INDEX--,//,15X,42HJ = 1/2 * INTEGRAL (Y°*{A}*Y + U°*{B}*U)DT,
     6//,20X,32HSTATE FEEDBACK GAIN DEFINITION--,//,25X,10HU = -{C}*X,//
     7,15X,45HDO YOU WISH TC CONTINUE?  TYPE "YES" OR "NO".)
900   FORMAT (25X,14H--DATA ENTRY--,//,5X,49HALTHOUGH OPTSYSX IS SPECIFI
     1CALLY DESIGNED TO READ,/,5X,48HALL MATRIX DATA INTERACTIVELY, SEVE
     2RAL ALTERNATE,/,5X,31HMETHODS ARE AVAILABLE TO USERS:,//,10X,43HME
     3THOD 1--THE F,G,H,D, AND "GAMMA" MATRICES,/,13X,37HMAY BE READ FRO
     4M SEPARATE DATA FILES.,//,10X,50HMETHOD 2--THE F,G,H,D, AND "GAMMA
     5" MATRICES MAY BE,/,13X,45HEXPLICITLY DEFINED WITHIN SUBROUTINE "S
     6ETUP".,//,10X,52H(NOTE: IN EITHER CASE, THE USER SHOULD OBTAIN A C
     7OPY,/,17X,34HOF THE PROGRAM LISTING AND EXAMINE,/,17X,39HTHE EXAMP
```

```
     8LES CONTAINED IN S/R "SETUP".),//,10X,45HDO YOU WISH TO CONTINUE?
     9 TYPE "YES" OR "NO".)
910   FORMAT (//,5X,46HDO YCU WISH TO INPUT THE "F", "G", AND "GAMMA",/,
     110X,40HMATRICES FROM SUBROUTINE "SETUP" IAW THE,/,10X,40HMETHOD DE
     2SCRIBED ON THE PREVIOUS SCREEN?,//,15X,19HTYPE "YES" OR "NO".)
570   FORMAT (25X,24HGENERAL OPTSYSX OPTIONS:,//,10X,35HOPTION 1 -- SYST
     1EM ANALYSIS WITHOUT,/,22X,35HOPEN-LOOP EIGENSYSTEM CALCULATIONS.,/
     2/,10X,42HOPTION 2 -- SYSTEM ANALYSIS WITH OPEN-LOOP,/,22X,25HEIGEN
     3SYSTEM CALCULATIONS.,//,10X,39HOPTION 3 -- OPEN-LOOP EIGENSYSTEM F
     4OUND,/,22X,23HAND PROGRAM TERMINATES.,/,22X,39H("F"-MATRIX ENTRY F
     5OLLOWS IMMEDIATELY.),//,10X,48HOPTION 4 -- MODAL DISTRIBUTION MATR
     6ICES COMPUTED,/,22X,37HWITHOUT FILTER OR REGULATOR SYNTHESIS,/,22X
     7,25HOR STEADY-STATE ANALYSIS.,//,15X,30HSELECT AN OPTION: 1,2,3, O
     8R 4.)
580   FORMAT (//,5X,46HDO YCU DESIRE RMS VALUES OF STATE AND CONTROL?,//
     1,10X,19HTYPE "YES" OR "NO".)
590   FORMAT (/,20X,30HOPTSYSX LQR/CLASSICAL OPTIONS:,//,10X,43HOPTION 1
     1 -- OPTIMAL FILTER AND/OR REGULATOR,/,22X,37HSYNTHESIS WITH NO EXT
     2ERNAL "C" OR "K",/,22X,13HMATRIX INPUT.,//,10X,43HOPTION 2 -- OPTI
     3MAL FILTER AND/OR REGULATOR,/,22X,27HSYNTHESIS WITH EXTERNAL "C",/
     4,22X,13HMATRIX INPUT.,//,10X,43HOPTION 3 -- OPTIMAL FILTER AND/OR
     5REGULATOR,/,22X,27HSYNTHESIS WITH EXTERNAL "K",/,22X,13HMATRIX INP
     6UT.,//,10X,  'OPTION 4 -- SYSTEM ANALYSIS'              ,/,22X,
     7          'WITH EXTERNAL "C" AND "K" MATRIX INPUT.',//,10X
     8,32HSELECT AN OPTION: 1, 2, 3, OR 4.)
600   FORMAT (//,5X,50HDO YCU WISH TO DETERMINE THE STEADY-STATE RESPONS
     1E,/,8X,27HFOR A CONSTANT DISTURBANCE?,//,10X,19HTYPE "YES" OR "NO"
     2.)
610   FORMAT (5X,47HDO YOU WISH TO DETERMINE THE MODAL DISTRIBUTION,/,8X
     1,18HAND GAIN MATRICES?,//,10X,19HTYPE "YES" OR "NO".)
620   FORMAT (//,5X,36HOPEN-LOOP TRANSFER FUNCTION CPTIONS:,//,10X,53HOP
     1TION 1 -- NO OPEN-LOOP TRANSFER FUNCTIONS COMPUTED.,//,10X,48HOPTI
     2ON 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.,//,10X,42HOPTION 3 --
     3 ONLY POLES AND ZEROS COMPUTED.,//,10X,45HCPTION 4 -- ONLY POLES A
     4ND RESIDUES COMPUTED.,//,10X,32HSELECT AN OPTION: 1, 2, 3, OR 4.,/
     5//,8X,32HTF ANALYSIS CAUSES MODAL OUTPUT,//,5X,61HPOLES AND ZEROS
     6 MUST BE SELECTED IF YOU WISH TO USE OPTGRAPH.)
630   FORMAT (//,5X,32HNOISE TRANSFER FUNCTION OPTIONS:,//,10X,49HOPTION
     1 1 -- NO NOISE TRANSFER FUNCTIONS COMPUTED.,//,10X,48HOPTION 2 --
     2POLES, RESIDUES, AND ZEROS COMPUTED.,//,10X,42HCPTION 3 -- ONLY PO
     3LES AND ZERCS COMPUTEL.,//,10X,45HOPTION 4 -- ONLY POLES AND RESID
     4UES COMPUTED.,//,10X,32HSELECT AN OPTION: 1, 2, 3, OR 4.)
640   FORMAT (//,5X,38HCOMPENSATOR TRANSFER FUNCTION OPTIONS:,//,10X,49H
     1OPTION 1 -- NO COMP. TRANSFER FUNCTIONS COMPUTED.,//,10X,48HOPTION
     2 2 -- POLES, RESIDUES, AND ZEROS COMPUTED.,//,10X,42HOPTION 3 -- O
     3NLY POLES AND ZEROS COMPUTED.,//,10X,45HOPTION 4 -- ONLY POLES AND
     4 RESIDUES COMPUTED.,//,15X,45H(NOTE: A COMPENSATOR TRANSFER FUNCTI
     5ON CAN BE,/,22X,33HCOMPUTED ONLY IF BOTH A REGULATOR,/,22X,26HAND
```

```
      6FILTER ARE SYNTHESIZED,/,22X,14HAND/OR INPUT.},//,10X,32HSELECT AN
      7 OPTION: 1, 2, 3, OR 4.)
650    FORMAT (//,5X,39HWILL A FEED-FORWARD DISTRIBUTION MATRIX,/,5X,25H{
      1"D" - MATRIX} BE INPUT ?,//,15X,19HTYPE "YES" OR "NO".)
660    FORMAT (/,5X,63H THIS OPTION DETERMINES THE CRITERIA FOR DECIDING
      1WHEN A MARKOV,/,8X,58HPARAMETER IS ZERO-THE MARKOV PARAMETER INDIC
      2ATES THE ORDER,/,8X,54HOF THE NUMERATOR POLYNCMIAL OF EACH TRANSFE
      3R FUNCTION.,//,8X,52HALL "N" ZEROS OF THIS POLYNOMIAL ARE PRINTED
      4OUT AND,/,8X,52HTHIS TEST TELLS HOW MANY EXTRA ROOTS EXIST AT Z =
      50.,/,8X,41HLESS THAN 10.0**{-IE} IS CONSIDERED ZERO.,//,8X,47H THE
      6 DEFAULT VALUE OF THIS PARAMETER {IE} IS 6.,/,8X,28HIN OTHER WORDS
      7,ESP = 1.0E-6.,//,10X,66HIF YOU DESIRE A DIFFERENT MARKOV CRITERIA
      8, TYPE THE INTEGER VALUE.,//,10X,48HIF YOU DESIRE THE DEFAULT VALU
      9E, TYPE "0" {ZERO})
670    FORMAT (//,5X,61HDO YOU DESIRE TO SYNTHESIZE A STABLE FILTER {OR R
      1EGULATOR} BY,/,8X,34HDESTABILIZING THE ORIGINAL SYSTEM?,//,12X,52H
      2{NOTE:WORKS FOR FILTER OR REGULATOR BUT NOT FOR BOTH,/,20X,17HIN T
      3HE SAME RUN.},//,10X,19HTYPE "YES" OR "NO".)
680    FORMAT (5X,53HDO YOU DESIRE TO PRINT THE EULER-LAGRANGE EIGENSYSTE
      1M,/,8X,50HPRIOR TO DECOMPOSITION {FOR CHECKING THE PROGRAM}?,//,10
      2X,19HTYPE "YES" OR "NC".)
690    FORMAT (//,5X,39HPOWER SPECTRAL DENSITY {PSD} OPTION 1 :,//,10X,53
      1HOPTION 1 -- COMPUTE THE PSD OF THE OUTPUTS AND/OR THE,/,22X,48HCO
      2NTROLS OF THE CONTROLLED SYSTEM WHEN FORCED BY,/,22X,45HPROCESS AN
      3D MEASUREMENT NOISE.  {NOTE: BOTH A,/,22X,46HREGULATOR AND A FILTE
      4R MUST BE RESIDENT IN THE,/,22X,28HPROGRAM TO USE THIS OPTION.},//
      5,10X,53HOPTION 2 -- SAME AS OPTION 1 ABOVE BUT ONLY PRINT THE,/,22
      6X,34HRESIDUES OF EACH TRANSFER FUNCTION,/,22X,28HUSED IN THE PSD C
      7OMPUTATION.,//,10X,24HOPTION 3 -- NOT DESIRED.,///,10X,29HSELECT A
      8N OPTION: 1, 2, OR 3.)
700    FORMAT (//,5X,39HPOWER SPECTRAL DENSITY {PSD} OPTION 2 :,//,10X,35
      1HOPTION 1 -- PSD OUTPUT NOT DESIRED.,//,10X,38HOPTION 2 -- COMPUTE
      2 ONLY OUTPUT PSD.,//,10X,39HOPTION 3 -- COMPUTE ONLY CONTROL PSD
      3.,//,10X,50HOPTION 4 -- COMPUTE BOTH OUTPUT AND CONTROL PSD.,/
      4/,15X,32HSELECT AN OPTION: 1, 2, 3, OR 4.)
710    FORMAT (//,5X,39HDO YCU DESIRE REGULATOR SYNTHESIS ONLY?,//,10X,19
      1HTYPE "YES" OR "NO".,//)
720    FORMAT (/,5X,47HENTER THE # OF STATES {NS} OF THE SYSTEM MATRIX,/,
      15X,13H{"F"-MATRIX}.)
730    FORMAT (/,5X,56HENTER THE # OF CONTROLS {NC} OF THE CONTROL SYSTEM
      1 MODEL,/,5X,13H{"G"-MATRIX}.)
740    FORMAT (/,5X,54HENTER THE # OF MEASUREMENTS OR OBSERVATIONS {NO} O
      1FTHE,/,5X,13H{"H"-MATRIX}.)
750    FORMAT (/,5X,48HENTER THE # OF PROCESS NOISE SOURCES {NG} OF THE,/
      1,5X,17H{"GAMMA"-MATRIX}.)
760    FORMAT (5X,52HFLAG/PARAMETER SETTINGS FOR THIS RUN ARE AS FOLLOWS:
      1,/)
770    FORMAT (1X,3HIOL,2X,2HIQ,2X,2HIR,2X,3HISS,2X,2HIM,2X,4HITF1,2X,4HI
```

```
     1TF2,2X,4HITF3,2X,5HIFLFW,2X,2HIE,2X,6HIDEBUG,2X,4HISET,2X,6HIDSTAB
     2,/)
780   FORMAT (1X,I2,3X,I2,3X,I2,2X,I2,3X,I2,3X,I2,4X,I2,4X,I2,4X,I2,4X,I
     12,3X,I2,6X,I2,5X,I2,/)
790   FORMAT (1X,4HIPSD,2X,3HIYU,2X,5HINORM,2X,4HIREG,2X,2HNS,2X,2HNC,2X
     1,3HNOB,2X,2HNG,/)
800   FORMAT (2X,I2,3X,I2,4X,I2,5X,I2,3X,I2,2X,I2,3X,I2,2X,I2,//)
810   FORMAT (2X,17HORDER OF SYSTEM =,I3,//,2X,20HNUMBER OF CONTROLS =,I
     13,//,2X,24HNUMBER OF CBSERVATIONS =,I3,//,2X,33HNUMBER OF PROCESS
     2NOISE SOURCES =,I3,////)
820   FORMAT (5X,53HDETERMINE THE NORMALIZATION PARAMETER (INORM) FOR TH
     1E,/,5X,55HPOWER SPECTRAL DENSITY (PSD) OPTION YOU HAVE PREVIOUSLY,
     2/,5X,52HCHOSEN. TWO PSD NORMALIZATION METHODS ARE AVAILABLE:,//,10
     3X,54HMETHOD 1 -- PSD IS NORMALIZED BY THE I-NORTH PROCESS,/,21X,
     429HNOISE MINUS "Q"(INCRM,INORM).,/,21X,49H(NOTE: "Q" IS AN OPTIMAL
     5 STATE WEIGHTING MATRIX.),/,21X,34HIN THIS METHOD, INORM =0,1,...,
     6NG.,//,10X,63HMETHOD 2 -- PSD IS NORMALIZED BY THE (INORM -NG)TH
     7MEASUREMENT,/,21X,39HNOISE MINUS "R"(INORM - NG,INORM - NG).,/,21X
     8,51H(NOTE: "R" IS AN OPTIMAL CONTROL WEIGHTING MATRIX.),/,21X,44HI
     9N THIS METHCD, INORM = NG + 1,...,NG + NOB.,//,10X,51HSELECT AN IN
      TEGER FROM 0 - 16 REPRESENTING YOUR PSD,/,15X,27HNORMALIZATION REQ
      UIREMENTS.,//,10X,53HIF PSD NORMALIZATION IS NOT DESIRED ENTER "0"
       (ZERO).)
830   FORMAT (5X,43HANALYSIS COMPLETE. DO YOU WANT ANOTHER RUN?,/,15X,19
     1HTYPE "YES" OR "NO".)
840   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "F"-MATRIX FROM THE LAST
     1,/,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
850   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "H"-MATRIX FROM THE LAST
     1,/,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
860   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "G"-MATRIX FROM THE LAST
     1,/,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
865   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "D"-MATRIX FROM THE LAST
     1,/,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
870   FORMAT (///,5X,52HDO YOU WISH TO SAVE THE "GAMMA"-MATRIX FROM THE
     1LAST,/,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?,//,5X,39HNOTE: T
     2HE MATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE
```

```
     3 INTERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27
     4HINDIVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
872   FORMAT (///,5X,'DO YOU WISH TO SAVE THE "A"-MATRIX FROM THE LAST'
     1  ,/,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?,//,5X,39HNOTE: T
     2HE MATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE
     3 INTERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27
     4HINDIVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
874   FORMAT (///,5X,'DO YOU WISH TO SAVE THE "B"-MATRIX FROM THE LAST'
     1  ,/,5X,36HRUN TO BE USED IN THE FOLLOWING RUN?,//,5X,39HNOTE: T
     2HE MATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE
     3 INTERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27
     4HINDIVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
880   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
920   FORMAT (//,41H........OPTSYSX IS NOW TERMINATED........,//)
C-----------------------------------------------------OPTGRAPH DATA-----------
1000  FORMAT(I1,I4,5I5)
C-----------------------------------------------------------------------------
      END
C=============================================================================
      SUBROUTINE SETUP (BA,G,GAM,HO,D,NS,NC,NG,NO)
C=============================================================================
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BA(NS,NS),G(NS,NC),GAM(NS,NG),HO(NO,NS),D(NO,NC)
      COMMON /PROG/ IOL,IO,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,IDEB
     1UG,ISET,IREG,IPSD,IYU,INORM
C-----------------------------------------------------------------------------
C   FILE DEFINITICNS
C-----------------------------------------------------------------------------
      CALL FRTCMS ('FILEDEF ','03      ','DISK    ','X16X16  ',
     1'DATA    ','A       ')
C-----------------------------------------------------------------------------
C   THIS IS AN EXAMPLE OF A 83 X 84 DATA FILE X29A83 DATA A1 READ FROM
C   A USER'S DISK AND CONVERTED (FROM A "DUMMY" ARRAY NAMED 'DUM') TO A
C   SYMMETRIC ARRAY.  THE FORMAT STATEMENT MUST MATCH YOUR DISK DATA
C   FORMAT OR THE PROGRAM WILL FAIL!  NOTE: ALL PROGRAM DIMENSIONS
C   MUST BE ENLARGED ACCORDINGLY FOR A SYSTEM OF THIS SIZE.
C-----------------------------------------------------------------------------
C     READ (3,100)
C     READ (3,100)
C     DO 20 I=1,NS
C     READ (3,50) (BA(I,J),J=1,NS)
C0    CONTINUE
C-----------------------------------------------------------------------------
C   THESE ARE EXAMPLES OF SEVERAL POSSIBLE METHCDS OF ARRAY GENERATION
C   WITHIN SUBROUTINE SETUP.THE "GAM" ARRAY WAS SET TO ZERO SINCE NO
C   "NOISE" WAS PRESENT, AND THE NON-ZERO ELEMENTS OF THE "G" ARRAY WERE
C   EXPLICITLY DEFINED. THEY COULD ALSO BE READ FROM FILES AS ABOVE.
C-----------------------------------------------------------------------------
```

```
      DO 40  I=1,NS
      DO 30  J=1,NC
      DO 25  K=1,NO
      GAM(I,J)=0.0D+00
      G(I,J) =0.0D+00
      D(K,J) =0.0D+00
 25   CONTINUE
 30   CONTINUE
 40   CONTINUE
C     G(82,1)= 0.1000D+00
C     G(52,1)= 0.362D+07
C     G(77,1)=-0.1591D+02
C     G(78,1)= 0.2448D+00
C     G(79,1)= 0.2448D+00
C     G(81,1)= 0.1000D+00
C-----------------------------------------------------------------------
C     READ (3,100)
C     DO 70 I=1,NS
C     READ (3,50) (G(I,J),J=1,NC)
C70   CONTINUE
C-----------------------------------------------------------------------
C  THIS IS AN EXAMPLE OF ONE POSSIBLE METHOD OF ARRAY GENERATION      =
C  WITHIN THE PROGRAM ITSELF.  FOR VERY LARGE DATA ARRAYS, THIS METHOD =
C  MAY BE PREFERABLE TO SOME USERS OVER INTERACTIVE ENTRY OF EACH     =
C  INDIVIDUAL ELEMENT.                                                =
C-----------------------------------------------------------------------
C     DO 2 I=1,11
C        DO 1 J=1,82
C            HO (I,J)    = 0.0D+00
C            HO (1,1)    = 0.11520D+00
C            HO (2,75)   = 0.5730D+02
C            HO (3,74)   = 0.1000D+01
C            HO (4,63)   = 0.5730D+02
C            HO (5,62)   = 0.1000D+01
C            HO (6,76)   = 0.5730D+02
C            HO (7,44)   = 0.5730D+02
C            HO (8,45)   = 0.5730D+02
C            HO (9,46)   = 0.5730D+02
C            HO (10,47)  = 0.5730D+02
C            HO (11,48)  = 0.5730D+02
C1       CONTINUE
C2    CONTINUE
C     GO TO 90
C3    CONTINUE
C-----------------------------------------------------------------------
C     READ (3,100)
C     DO 90 I=1,NO
C     READ (3,50) (HO(I,J),J=1,NS)
```

```
C90    CONTINUE
       DO 1 I=1,NS
 1     READ (3,50) (BA(I,J),J=1,NS)
       READ (3,50) ((G(I,J),J=1,NC),I=1,NS)
       READ (3,50) ((HO(I,J),J=1,NS),I=1,NO)
       READ (3,50) ((D(I,J),J=1,NC),I=1,NO)
       RETURN
C-----------------------------------------------------------------------
 50    FORMAT (6E12.5)
C50    FORMAT (5E12.4)
C50    FORMAT (3(D22.15,3X))
 100   FORMAT ()
       END
C=======================================================================
       SUBROUTINE CHECK (EPS,NC,NG,NO,IRET)
C      CHECKS THE CONSISTENCY OF REQUESTED OPTIONS.                    =
C=======================================================================
       DOUBLE PRECISION EPS
       COMMON /PROG/ IOL,IO,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,IDEB
      1UG,ISET,IREG,IPSD,IYU,INORM
C-------SET MODAL ANALYSIS WHEN OL EIGENSYS OR OL TF REQUESTED----------
       IF (IM .EQ. 1 .AND. IOL .EQ. 0) IOL=1
       IF (IOL .EQ. 3 .OR. ITF1 .NE. 0) IM=1
C----------------CHECK TO SEE IF H MATRIX INPUT-------------------------
       IF (NO .NE. 0 .OR. IOI .GE. 2) GO TO 10
       WRITE (5,90)
       IRET=1
       RETURN
10     CONTINUE
C----------------TRANSFER FUNCTION CHECKS-------------------------------
       IF (IE .EQ. 0) IE=6
       EPS=10.**(-IE)
C---------------------OPEN LOOP TF--------------------------------------
       IF (ITF1 .EQ. 0 .OR. NC .NE . 0) GO TO 20
       WRITE (5,100)
       IRET=1
       RETURN
C---------------------COMPENSATOR TF------------------------------------
20     IF (ITF3 .EQ. 0) GO TC 30
       IF (IREG .EQ. 0 .AND. (NC .NE. 0 .AND. NG .NE. 0)) GO TO 30
       WRITE (5,110)
       IRET=1
       RETURN
30     CONTINUE
C---------------------NOISE TF------------------------------------------
       IF (ITF2 .EQ. 0) GO TC 40
       IF (NG.NE. 0 .AND. NC .NE. 0) GO TO 40
       WRITE (5,120)
```

```
      IRET=1
      RETURN
C-----------------DESTABILIZATION RESTRICTIONS------------------------
40    IF (IDSTAB .EQ. 0) GO TO 50
      IF (NC .EQ. 0) GO TO 50
      IF (NG .NE. 0) IREG=1
      WRITE (5,130)
      IF (IREG .EQ. 1) GO TO 50
      IRET=1
      RETURN
50    CONTINUE
C-----------------PSD INPUT-------------------------------------------
      IF (IPSD .EQ. 0) GO TC 80
      IF (IPSD .LT. 0 .OR. IPSD .GT. 3) GO TO 60
      IF (IYU .LT. 0 .OR. IYU .GT. 3) GO TO 60
      IF (INORM .LT. 0 .OR. INORM .GT. NG+NO) GO TO 60
      GO TO 70
60    WRITE (5,140)
      IRET=1
      RETURN
70    IF (IREG .EQ. 0 .AND. NC .NE. 0) GO TO 80
      WRITE (5,150)
      IRET=1
      RETURN
80    CONTINUE
      RETURN
C---------------------------------------------------------------------
90    FORMAT (//,5X,49H H - MATRIX MUST BE INPUT, I.E. "NO" MUST BE > 0.
     1,//)
100   FORMAT (//,5X,46H(G) MATRIX MUST BE INPUT, I.E. NC MUST BE > 0.,/,
     110X,26HTO CCMPUTE OPEN LOOP T. F.,//)
110   FORMAT (//,5X,48HREGUIATOR AND FILTER SYNTHESIS MUST BE REQUESTED,
     1/,5X,44HIN THE SAME RUN TO COMPUTE COMPENSATOR T. F.,/,5X,47HI.E.
     2IREG MUST = 0.; "NC" AND "NG" MUST BE > 0.,//)
120   FORMAT (//,5X,51HNOISE T. F. CALCULATED ONLY WHEN REGULATOR DESIGN
     1ED,/,5X,47HI.E. IREG MUST = 1.; "NC" AND "NG" MUST BE > 0.,//)
130   FORMAT (//,5X,47HDESTABILIZATION OPTION DESIGNED FOR A REGULATOR,/
     1,5X,38HOR FILTER BUT NOT BOTH SIMULTANEOUSLY.,//,5X,55HIF "NG" > 0
     2. THE REGULATOR OPTION IS AUTOMATICALLY SET!,//)
140   FORMAT (//,5X,49H ********** INCONSISTENT PSD INPUT FLAGS ********
     1,//)
150   FORMAT (//,5X,44HBOTH A REGULATOR AND FILTER MUST BE RESIDENT,/,10
     1X,42HTO COMPUTE THE PSD OF A CONTROLLED SYSTEM!,/,10X,42HI.E. IREG
     2 MUST BE 0. AND "NC" MUST BE > 0.,//)
      END
C=====================================================================
      SUBROUTINE INNER (NS,NC,NO,NG,N2,ACL,B,BA,CI,CR,CQ,CWI,CWR,D,FBGC,
     1FBGE,G,GAM,GM,GN,HO,D1,D2,PRO,RM,RC,Q,SC,WR,WI,W11,W21,X,WNORM,WNO
```

```
1480  FORMAT (//,25X,28H....DESTABILIZATION CASE....,//,10X,39HTHE FOLLO
     1WING VALUES WILL BE ADDED DOWN,/,10X,49HTHE DIAGONAL OF THE "F" MA
     2TRIX TO DESTABILIZE IT.,/,10X,41HOPTIMAL GAINS FOR THE DESTABILIZE
     3D SYSTEM,/,10X,39HARE THEN USED AS FIXED SUBOPTIMAL GAINS,/,10X,28
     4HFOR THE SYSTEM CALCULATIONS.,//)
1490  FORMAT (///43H PROGRAM TERMINATING DUE TO UNSTABLE SYSTEM)
1500  FORMAT (//,5X,31HOPEN LOOP TRANSFER FUNCTIONS...)
1510  FORMAT (//,5X,32H EULER-LAGRANGE SYSTEM MATRIX...,//)
1520  FORMAT (//,5X,43HEIGENVALUES AND EIGENVECTORS OF THE 2N X 2N,/,5X,
     145HEULER-LAGRANGE SYSTEM AFTER HQR2.............,//)
1530  FORMAT (1X,1P2D13.6)
1540  FORMAT (1X)
1550  FORMAT (//,5X,41HEIGENSYSTEM OF OPTIMAL REGULATOR..........,//)
1560  FORMAT (//,5X,41HEIGENSYSTEM OF OPTIMAL ESTIMATOR..........,//)
1570  FORMAT (//,5X,39H EIGENVECTORS FROM RGAIN PRIOR TO CNORM,//)
1580  FORMAT (//,2X,57HTHE OPTIMAL FEEDBACK GAIN CONTROL MATRIX...C=BINV
     1*GT*S....,//)
1590  FORMAT (10(2X,1PD11.4))
1600  FORMAT (//,5X,45HTHE CLOSED LOOP DYNAMICS MATRIX ......F-G*C..,//)
1610  FORMAT (///,60H   PROGRAM TERMINATING DUE TO UNSTABLE CLOSED LOOP
     1    SYSTEM)
1620  FORMAT (//,2X,'NOISE TRANSFER FUNCTIONS THROUGH THE CLOSED-LOOP SY
     1STEM..',//)
1630  FORMAT (//,5X,45HFILTER STEADY STATE GAINS................K...,//)
1640  FORMAT (1X,2X,1P6D14.6)
1650  FORMAT (//,5X,45HTHE CLOSED LOOP FILTER DYNAMICS MATRIX IS....,//)
1660  FORMAT (////,43H PROGRAM TERMINATING DUE TO UNSTABLE FILTER)
1670  FORMAT (//,5X,45HTHE COVARIANCE OF THE ESTIMATION ERROR....P..,//)
1680  FORMAT (//,5X,45HRMS VALUES OF THE ESTIMATION ERROR...........,//)
1690  FORMAT ((5(1X,1PD13.6)))
1700  FORMAT (//,5X,45HTHE COVARIANCE OF THE ESTIMATE.........XHAT..,//)
1710  FORMAT (//,5X,45HTHE STATE COVARIANCE MATRIX.....X=XHAT + P...,//)
1720  FORMAT (//,5X,45HTHE CONTROL COVARIANCE.........U=C*XHAT*CT...,//)
1730  FORMAT (1P6D14.6)
1740  FORMAT (//,2X,18HSTATE RMS RESPONSE,20X,20HCONTROL RMS RESPONSE,/)
1750  FORMAT (1X,1PD15.7,25X,D15.7)
1760  FORMAT (//,5X,50HCOMPENSATOR TRANSFER FUNCTIONS FROM MEAS. TO INPU
     1T,/,5X,52H.................U/Z = -C*(SI-F+G*C+K*H)INV*K...,//)
1770  FORMAT (//,5X,46HSTEADY DISTURBANCE VECTOR..................W..,//
     1,10(1X,1PD12.4/))
1780  FORMAT (//,5X,45HSTEADY STATE VALUES OF STATE VAR. ARE........,//)
1790  FORMAT (//,5X,47HSTEADY STATE CONTROL IS .......................,/
     1/10(1X,1PD12.4/))
1800  FORMAT (//,5X,49HENTER THE MAGNITUDE OF THE DESTABILIZATION VECTOR
     1,/,8X,47HTO BE ADDED DOWN THE DIAGONAL OF THE "F"-MATRIX,/,8X,18HT
     2O DESTABILIZE IT.,//)
C------------------------------------------------------OPGRAPH DATA-----------
2000  FORMAT(5X,2D30.14)
```

```
1290  CONTINUE
C----------COMPUTE PSD FUNCTIONS OF THE CONTROLLED SYSTEM--------------
      IF (IPSD .EQ. 0) GO TC 1310
      IF (IYU .LT. 3) GO TO 1300
      CALL PSDCAL (M,NS,RM,X,NC,GW,GV,FBGC,NO,HY,HU,HO,FBGE,NG,
     1 GAM,ACL,BA,WR,WI,D1,L2,JCF,RES,Q,RC,BB,CC,1   ,IPSD,INORM)
      CALL PSDCAL (M,NS,RM,X,NC,GW,GV,FBGC,NO,HY,HY,HO,FBGE,NG,
     1 GAM,ACL,BA,WR,WI,D1,L2,JCF,RES,Q,RC,BB,CC,2   ,IPSD,INORM)
      GO TO 1310
1300  CALL PSDCAL (M,NS,RM,X,NC,GW,GV,FBGC,NO,HY,HU,HO,FBGE,NG,
     1 GAM,ACL,BA,WR,WI,D1,L2,JCF,RES,Q,RC,BB,CC,IYU,IPSD,INORM)
1310  IF (ISS .EQ. 0) RETURN
      IF (NC .NE. 0) GO TO 1330
      DO 1320 I=1,NS
      DO 1320 J=1,NS
1320  ACL(I,J)=BA(I,J)
1330  CONTINUE
      CALL MINV (NSQ,ACL,NS,DDD,D1,D2)
      CALL READW (NG,WR)
      WRITE (6,1770) (WR(I),I=1,NG)
      WRITE (6,1780)
      DO 1340 I=1,NS
      WI(I)=0.0
      DO 1340 J=1,NG
1340  WI(I)=WI(I)+GAM(I,J)*WR(J)
      DO 1360 I=1,NS
      CR(I)=0.0
      DO 1350 J=1,NS
1350  CR(I)=CR(I)-ACL(I,J)*WI(J)
1360  WRITE (6,1390) CR(I)
      DO 1370 I=1,NC
      CI(I)=0.0
      DO 1370 J=1,NS
1370  CI(I)=CI(I)+FBGC(I,J)*CR(J)
      WRITE (6,1790) (CI(I),I=1,NC)
      RETURN
C----------------------------------------------------------------------
C670  FORMAT (2X,1P6D14.6,/,2X,6D14.6)
1380  FORMAT (//,5X,45HOPEN LOOP DYNAMICS MATRIX.................F..,//)
1390  FORMAT (10(2X,0PD11.4))
1400  FORMAT (//,5X,45HTHE CONTROL DISTRIBUTION MATRIX...........G..,//)
1410  FORMAT (//,5X,45HTHE CONTROL COST MATRIX...................B..,//)
1420  FORMAT (//,5X,45HPROCESS NOISE DISTRIBUTION MATRIX.....GAMMA..,//)
1430  FORMAT (//,5X,45HPOWER SPECTRAL DENSITY - PROCESS NOISE....Q..,//)
1440  FORMAT (//,5X,45HMEASUREMENT SCALING MATRIX................H..,//)
1450  FORMAT (//,5X,45HPOWER SPECTRAL DENSITY-MEASUREMENT NOISE..R..,//)
1460  FORMAT (//,5X,45HOUTPUT COST MATRIX........................A..,//)
1470  FORMAT (//,5X,45HMEASUREMENT FEEDTHROUGH MATRIX............D..,//)
```

```
 1120   CONTINUE
        DO 1140 I=1,NC
        DO 1140 J=1,NC
        SC(I,J)=0.D0
        DO 1130 K=1,NS
 1130   SC(I,J)=SC(I,J)+FBGC(I,K)*PRO(K,J)
 1140   CONTINUE
 1150   IF (IREG .EQ. 0) GO TC 1170
        DO 1160 I=1,NS
        DO 1160 J=1,NS
 1160   CQ(I,J)=GM(I,J)
        GO TO 1190
 1170   WRITE (6,1700)
        CALL RAPRNT (MH,MH,MH,5,GM,4,'(5(1X,1PD13.6))')
        IF (IR.GT.2) GO TO 1150
        DO 1180 I=1,MH
        DO 1180 J=1,MH
 1180   CQ(I,J)=GN(I,J)+GM(I,J)
 1190   CONTINUE
        WRITE (6,1710)
        CALL RAPRNT (MH,MH,MH,5,CQ,4,'(5(1X,1PD13.6))')
        IF (NC.EQ.0) GO TO 1210
        WRITE (6,1720)
        DO 1200 I=1,NC
 1200   WRITE (6,1730) (SC(I,J),J=1,NC)
 1210   DO 1220 I=1,NS
 1220   CQ(I,I)=DSQRT(CQ(I,I))
        IF (NC.EQ.0) GO TO 1240
        DO 1230 I=1,NC
 1230   SC(I,I)=DSQRT(SC(I,I))
 1240   WRITE (6,1740)
        DO 1250 I=1,NS
        IF (I.LE.NC) WRITE (6,1750) CQ(I,I),SC(I,I)
        IF (I.GT.NC) WRITE (6,1750) CQ(I,I)
 1250   CONTINUE
 1260   IF (ITF3 .EC. 0) GO TC 1290
C-------FORM COMPENSATOR FRCM MEAS TO INPUT AND COMPUTE TF-------------
        DO 1280 I=1,NS
        DO 1280 J=1,NS
        SUM=0.D0
        DO 1270 K=1,NO
 1270   SUM=SUM+FBGE(I,K)*HO(K,J)
 1280   CQ(I,J)=ACL(I,J)-SUM
        WRITE (6,1760)
        ITFX=3
        IZERO=0
        CALL TF (NS,NS,NSQ,CQ,AA,NO,FBGE,BM,NC,FBGC,CM,IZERO,D,BB,CC,CP,
       1WR,WI,CWR,CWI,SC,JCF,RES,D1,D2,DDD,EPS,ITF3,ITFX)
```

```
      DO 980 I=1,MH
980   X(I,I)=DSQRT(GN(I,I))
      WRITE (6,1690) (X(I,I),I=1,MH)
      IF (IQ.EQ.0) GO TO 1260
      DO 1000 I=1,NC
      DO 1000 J=1,NS
      SUM=0.0
      DO 990 K=1,NS
990   SUM=SUM+FBGC(I,K)*GN(K,J)
1000  X(I,J)=SUM
      DO 1020 I=1,NS
      DO 1020 J=1,NS
      SUM=0.0
      IF (NC.EQ.0) GO TO 1020
      DO 1010 K=1,NC
1010  SUM=SUM+G(I,K)*X(K,J)
1020  PRO(I,J)=CQ(I,J)+SUM
      CALL SCOV (NS,SC,W11,CWR,CWI,NS,GM,W21,CR,CI,PRO,BA)
      IF (NC.EQ.0) GO TO 1040
      DO 1030 I=1,NC
      DO 1030 J=1,NS
      W21(I,J)=0.0
      DO 1030 K=1,NS
1030  W21(I,J)=W21(I,J)+FBGC(I,K)*BA(J,K)
1040  DO 1060 I=1,NS
      DO 1060 J=1,NS
      SUM=0.0
      IF (NC.EQ.0) GO TO 1060
      DO 1050 K=1,NC
1050  SUM=SUM+G(I,K)*W21(K,J)
1060  PRO(I,J)=SUM
      DO 1070 I=1,NS
      DO 1070 J=I,NS
      PRO(I,J)=PRO(I,J)+CQ(I,J)+PRO(J,I)
1070  PRO(J,I)=PRO(I,J)
      CALL SCOV (NS,SC,W11,CWR,CWI,NS,SC,W11,CWR,CWI,PRO,CQ)
      DO 1080 I=1,NS
      DO 1080 J=I,NS
      GM(I,J)=CQ(I,J)-BA(I,J)-BA(J,I)+GN(I,J)
1080  GM(J,I)=GM(I,J)
      GO TO 1100
1090  CALL SCOV (NS,SC,W11,CWR,CWI,NS,SC,W11,CWR,CWI,CQ,GM)
1100  IF (NC.EQ.0) GO TO 1150
      DO 1120 I=1,NS
      DO 1120 J=1,NC
      PRO(I,J)=0.D0
      DO 1110 K=1,NS
1110  PRO(I,J)=PRO(I,J)+GM(I,K)*FBGC(J,K)
```

```
      CALL BALBAK (NS,NS,LOW,IHIGH,D1,NS,GM)
      WRITE (6,1560)
C-----------NORMALIZE AND PRINT SUBOPT. ESTIMATOR EIGENSYSTEM-----------
      IWRITE=5
      CALL CNORM (CR,CI,GM,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNORMI,HO,AA,
     1NO,NS)
      DO 880 I=1,NS
      IF (CR(I).LT.0.0) GO TO 880
      WRITE (5,1660)
      RETURN
880   CONTINUE
      GO TO 900
890   IF (IQ.EQ.0) GO TO 1260
900   DO 910 I=1,NO
      DO 910 J=1,MH
      PRO(I,J)=0.D0
      DO 910 K=1,NO
910   PRO(I,J)=PRC(I,J)+RC(I,K)*FBGE(J,K)
      DO 920 I=1,MH
      DO 920 J=1,MH
      CQ(I,J)=0.D0
      DO 920 K=1,NO
920   CQ(I,J)=CQ(I,J)-FBGE(I,K)*PRO(K,J)
930   CONTINUE
C-----------------THE RMS STATE AND CONTROL RESPONSES----------------
      IR=IR+1
      GO TO (1090,1090,940,940), IR
940   DO 950 I=1,NS
      DO 950 J=1,NG
      X(I,J)=0.0
      DO 950 K=1,NG
950   X(I,J)=X(I,J)+GAM(I,K)*Q(K,J)
      DO 970 I=1,NS
      DO 970 J=I,NS
      SUM=0.0
      DO 960 K=1,NG
960   SUM=SUM-X(I,K)*GAM(J,K)
      PRO(I,J)=SUM+CQ(I,J)
      PRO(J,I)=PRO(I,J)
      CQ(I,J)=SUM
      CQ(J,I)=SUM
      W21(I,J)=GM(I,J)
970   W21(J,I)=GM(J,I)
      CALL MINV (NSQ,W21,NS,DDD,D1,D2)
      CALL SCOV (NS,GM,W21,CR,CI,NS,GM,W21,CR,CI,PRO,GN)
      WRITE (6,1670)
      CALL RAPRNT (MH,MH,MH,5,GN,4,'(5(1X,1PD13.6))')
      WRITE (6,1680)
```

```
      IF (IDSTAB .EQ. 1) GO TO 770
C-----------NORMALIZE AND PRINT OPT. ESTIMATOR EIGENSYSTEM-------------
      IWRITE=4
      CALL CNORM (CR,CI,PRO,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNORMI,HO,AA,
     1NO,NS)
770   DO 780 I=1,MH
      DO 780 J=1,NO
780   PRO(I,J)=+HO(J,I)/RC(J,J)
      DO 790 I=1,MH
      DO 790 J=1,NO
      FBGE(I,J)=0.D0
      DO 790 K=1,MH
790   FBGE(I,J)=FBGE(I,J)+GN(I,K)*PRO(K,J)
      IF (IDSTAB .EQ. 1) GO TO 810
      WRITE (6,1670)
      CALL RAPRNT (MH,MH,MH,5,GN,4,'(5(1X,1PD13.6))')
      WRITE (6,1680)
      DO 800 I=1,MH
800   X(I,I)=DSQRT(GN(I,I))
      WRITE (6,1690)  (X(I,I),I=1,MH)
810   WRITE (6,1630)
      DO 820 I=1,MH
820   WRITE (6,1640)  (FBGE(I,J),J=1,NO)
C--------COMPUTE MODAL K MATRIX {OPEN LOOP U-INV SAVED IN WNORMI}-------
      IF (IM .NE. 1) GO TO 830
      CALL MODE (WNORMI,FBGE,AA,MH,MH,NO,4)
830   CONTINUE
C------RESET FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-------
      IF (IDSTAB .EQ. 0) GO TO 850
      DO 840 I=1,NS
      DO 840 J=1,NS
840   BA(I,J)=BA(I,J)-DSTORE(I,J)
      IR=2
850   CONTINUE
      DO 870 I=1,NS
      DO 870 J=1,NS
      SUM=0.0
      DO 860 K=1,NO
860   SUM=SUM+FBGE(I,K)*HO(K,J)
870   PRO(I,J)=BA(I,J)-SUM
      WRITE (6,1650)
      CALL RAPRNT (NS,NS,NS,5,PRO,4,'(5(1X,1PD13.6))')
      IF (IR .LT. 2) GO TO 890
      CALL BALANC (NS,NS,PRC,LOW,IHIGH,D1)
      CALL ORTHES (NS,NS,LOW,IHIGH,PRO,D2)
      CALL ORTRAN (NS,NS,LOW,IHIGH,PRO,D2,GM)
      CALL HQR2 (NS,NS,LOW,IHIGH,PRO,CR,CI,GM,IERR)
      IF (IERR .NE. 0) CALL EREXIT (NS,PRO,IERR)
```

```
C                                             RCOVARIANCE                =
C                                          ***R IS NOXNO MEASUREMENT NOISE  =
C                                             RCOVARIANCE                =
C   -HOT*RIN*HO      -FT                  ***HO IS NOXNS MEASUREMENT MATRIX=
C    ---                     ---            ***GM IS NSXNG STATE DISTURBANCE =
C                                              DISTRIBUTION MATRIX          =
C-------------------------------------------------------------------------------
      CALL READR (NO,RC)
      WRITE (6,1450)
      DO 680 I=1,NO
680   WRITE (6,1390) (RC(I,J),J=1,NO)
690   IF (ITF2 .EQ. 0) GO TC 700
C--------------------NOISE TRANSFER FUNCTIONS--------------------------
      WRITE (6,1620)
      ITFX=2
      IZERO=0
      CALL TF (NS,NS,NSQ,ACL,AA,NG,GAM,BM,NO,HO,CM,IZERO,D,BB,CC,CP,WR,
     1WI,CWR,CWI,SC,JCF,RES,D1,D2,DDD,EPS,ITF2,ITFX)
      IF (IREG .EQ. 1) RETURN
700   CONTINUE
      IF (IREG .EQ. 1) GO TC 930
      IF (IR.LT.2) GO TO 710
      CALL READFE (NS,NO,FBGE)
      GO TO 810
710   CONTINUE
C-----------THE MEASUREMENT MATRIX (HOT*RIN*HO===>SC------------------
      DO 720 I=1,NO
      DO 720 J=1,MH
720   PRO(I,J)=HO(I,J)/RC(I,I)
      DO 730 I=1,MH
      DO 730 J=1,MH
      RM(I+MH,J)=0.D0
      DO 730 K=1,NO
730   RM(I+MH,J)=RM(I+MH,J)-HO(K,I)*PRO(K,J)
C--------------------------GM*Q*GMT===>CC-----------------------------
      DO 740 I=1,NS
      DO 740 J=1,NS
      RM(I,J)=BA(I,J)
      RM(I+MH,J+MH)=-BA(J,I)
740   RM(I,J+NS)=CQ(I,J)
      GO TO 400
C-GO BACK TO 450 TO SET UP THE FILTER HAMILTONIAN; CALC.THE FILTER GAINS
750   CALL RGAIN (M,NS,NC,NCB,WR,WI,X,GN,GM ,RM,W21,D1,CR,CI,PRO,MHS,D2)
C--------------------CHECK EIGENVECTORS------------------------------
      IF (IDEBUG .EQ. 0) GO TO 760
      WRITE (6,1570)
      CALL RAPRNT (NS,NS,NS,9,PRO,4,'(9(1X,1PD13.6))')
760   CONTINUE
```

```
C-------NORMALIZE AND PRINT CLOSED LOOP SUBOPT. REG. EIGENSYSTEM--------
      IWRITE=3
      CALL CNORM (CWR,CWI,SC,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNORMI,FBGC,
     1AA,NC,NS)
      DO 570 I=1,NS
      IF (CWR(I).LT.0.0) GO TO 570
      WRITE (5,1610)
      RETURN
570   CONTINUE
      IF (IQ.NE.1) GO TO 590
      DO 580 I=1,NS
      DO 580 J=1,NS
580   W11(I,J)=SC(I,J)
      CALL MINV (NSQ,W11,NS,DDD,D1,D2)
590   NOB=NO
      IF ( NG .EQ. 0) RETURN
600   IF (ISET.EQ.1) GO TO 610
      CALL READG2 (NS,NG,IGAM,GAM)
610   CONTINUE
      IF (IOL .EQ. 3) GO TO 620
      CALL READQ (NG,Q)
620   WRITE (6,1420)
      DO 630 I=1,NS
630   WRITE (6,1390) (GAM(I,J),J=1,NG)
      IF (IM.NE. 1) GO TO 640
      CALL MODE (WNORMI,GAM,AA,NS,NS,NG,1)
640   CONTINUE
      IF (IOL .EQ. 3) RETURN
      WRITE (6,1430)
      DO 650 I=1,NG
650   WRITE (6,1390) (Q(I,J),J=1,NG)
      IF ((IQ.EQ.0).AND.(NG.EQ.0)) GO TO 1260
      DO 660 I=1,NG
      DO 660 J=1,NS
      PRO(I,J)=0.D0
      DO 660 K=1,NG
660   PRO(I,J)=PRO(I,J)+Q(I,K)*GAM(J,K)
      DO 670 I=1,NS
      DO 670 J=1,NS
      CQ(I,J)=0.D0
      DO 670 K=1,NG
670   CQ(I,J)=CQ(I,J)-GAM(I,K)*PRO(K,J)
      IF (IREG .EQ. 1) GO TO 690
C---CALCULATION OF FILTER GAINS:FORMATION OF ESTIMATION HAMILTONIAN-----
C                                                                      =
C    ___                 ___              ***F AND FT ARE SAME AS FOR  =
C                                           CONTROL HAMILTONIAN        =
C        F           -GM*Q*GMT          ***Q IS NGXNG STATE DISTURBANCE =
```

```
C------------FEEDBACK GAINS---> {U = -(BINVERSE)*GT*GN}-----------------
C-----------------------CALCULATE GT------------------------------------
      DO 490 I=1,NC
      DO 490 J=1,NS
      PRO(I,J)=0.D0
      DO 480 K=1,MH
480   PRO(I,J)=PRO(I,J)+G(K,I)*GN(K,J)
490   FBGC(I,J)=-PRO(I,J)/B(I,I)
      IF (IDSTAB .EQ. 1) GO TO 500
C---------NORMALIZE AND PRINT OPT. REG. CLOSED LOOP EIGENSYSTEM----------
      IWRITE=2
      CALL CNORM (CWR,CWI,SC,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNORMI,FBGC,
     ,AA,NC,NS)
C----------------THE OPTIMUM FEEDBACK CONTROL GAINS--------------------
500   WRITE (6,1580)
      DO 510 I=1,NC
510   WRITE (6,1590) (FBGC(I,J),J=1,NS)
C------COMPUTE MODAL C MATRIX {OPEN LOOP U-INVERSE SAVED IN WNORMI}-----
      IF (IM .NE. 1) GO TO 530
C-----------------------------------------------------------------------
C IN COMPUTING MODAL C RECOMPUTE U OPEN LOOP SINCE WNORM USED TO STORE
C U & U-INV FOR CLOSED LOOP SYSTEMS; WNORMI USED TO SAVE U-INV OPEN LOOP
C-----------------------------------------------------------------------
      DO 520 I=1,NS
      DO 520 J=1,NS
520   WNORM(I,J)=WNORMI(I,J)
      CALL MINV (NSQ,WNORM,NS,DDD,D1,D2)
      CALL MODE (WNORM,FBGC,AA,NS,NC,NS,3)
530   CONTINUE
C-----------------THE CLOSED LOOP DYNAMICS MATRIX-----------------------
      DO 550 I=1,NS
      DO 550 J=1,NS
      SUM=0.D0
      DO 540 K=1,NC
540   SUM=SUM+G(I,K)*FBGC(K,J)
550   ACL(I,J)=BA(I,J)+SUM
      WRITE (6,1600)
      CALL RAPRNT (MH,MH,MH,5,ACL,4,'(5(1X,1PD13.6))')
      IF (IR.NE.1.AND.IR.NE.3) GO TO 590
      DO 560 I=1,NS
      DO 560 J=1,NS
560   GN(I,J)=ACL(I,J)
      CALL BALANC (NS,NS,GN,LOW,IHIGH,D1)
      CALL ORTHES (NS,NS,LOW,IHIGH,GN,D2)
      CALL ORTRAN (NS,NS,LOW,IHIGH,GN,D2,SC)
      CALL HQR2 (NS,NS,LOW,IHIGH,GN,CWR,CWI,SC,IERR)
      IF (IERR .NE. 0) CALL EREXIT (NS,GN,IERR)
      CALL BALBAK (NS,NS,LOW,IHIGH,D1,NS,SC)
```

```
      RM(I,J+MH)=0.D0
      DO 380 K=1,NC
380   RM(I,J+MH)=RM(I,J+MH)-G(I,K)*PRO(K,J)
C-----------------2NX2N HAMILTONIAN MATRIX---------------------------
C---------------DIAGONAL BLOCKS---M11 AND M22------------------------
      DO 390 I=1,MH
      DO 390 J=1,MH
      RM(I,J)=BA(I,J)
      RM(I+MH,J+MH)=-BA(J,I)
C-----------------------------M21 BLOCK------------------------------
390   RM(I+MH,J)=-RM(I+MH,J)
C---------------M12 BLOCK IS DEFINED IN LINE 430 ABOVE---------------
400   CONTINUE
      IF (IDEBUG.EQ.0) GO TO 410
      WRITE (6,1510)
      CALL RAPRNT (M,M,M,9,RM,4,'(9(1X,1PD13.6))')
410   CALL BALANC (M,M,RM,LOW,IHIGH,D1)
      CALL ORTHES (M,M,LOW,IHIGH,RM,D2)
      CALL ORTRAN (M,M,LOW,IHIGH,RM,D2,X)
      CALL HQR2 (M,M,LOW,IHIGH,RM,WR,WI,X,IERR)
      IF (IERR .NE. 0) CALL EREXIT (M,RM,IERR)
      CALL BALBAK (M,M,LOW,IHIGH,D1,M,X)
C------------DEBUG DIAGNOSTICS ON EULER-LAGRANGE EQUATIONS-----------
      IF (IDEBUG .EQ. 0) GO TO 430
      WRITE (6,1520)
      DO 420 I=1,M
420   WRITE (6,1530) WR(I),WI(I)
      WRITE (6,1540)
      CALL RAPRNT (M,M,M,9,X,4,'(9(1X,1PD13.6))')
430   CONTINUE
      IF (IDSTAB .EQ. 1) GO TO 440
      IF (NOB.EQ.0) WRITE (6,1550)
      IF (NOB.NE.0) WRITE (6,1560)
440   IF (NOB.NE. 0) GO TO 750
      CALL RGAIN (M,NS,NC,NCB,WR,WI,X,GN,W11,RM,W21,D1,CWR,CWI,SC,MHS,D2
     1)
C-----------------------CHECK EIGENVECTORS---------------------------
      IF (IDEBUG .EQ. 0) GO TO 450
      WRITE (6,1570)
      CALL RAPRNT (NS,NS,NS,9,SC,4,'(9(1X,1PD13.6))')
450   CONTINUE
C------RESET FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-------
      IF (IDSTAB .EQ. 0) GO TO 470
      DO 460 I=1,NS
460   BA(I,I)=BA(I,I)-DESTAB(I)
      IR=1
470   CONTINUE
C------------------CALCULATION OF FEEDBACK GAIN----------------------
```

```
250    WRITE (6,1390)  (AY(I,J),J=1,NO)
       DO 260 I=1,NS
       DO 260 J=1,NS
       DO 260 K=1,NO
260    RM(I+MH,J)=RM(I+MH,J)+AA(K,I)*HO(K,J)
270    IF (ISET.EQ.1) GO TO 280
       CALL READG (NS,NC,ISAG,G)
280    CONTINUE
       IF (IOL.EQ.3) GO TO 290
       CALL READB (NC,ISAB,B)
290    WRITE (6,1400)
       DO 300 I=1,NS
300    WRITE (6,1390)  (G(I,J),J=1,NC)
       IF (IM.NE.1) GO TO 310
       CALL MODE (WNORMI,G,BM,NS,NS,NC,0)
310    CONTINUE
       IF (IOL.EQ.3) GO TO 340
       WRITE (6,1410)
       DO 320 I=1,NC
320    WRITE (6,1390)  (B(I,J),J=1,NC)
330    IF (ITF1.EQ.0) GO TO 350
C---------------OPEN LOOP TRANSFER FUNCTIONS-----------------------
340    WRITE (6,1500)
       ITFX=1
       CALL TF (NS,NS,NSQ,BA,AA,NC,G,BM,NO,HO,CM,IFDFW,D,BB,CC,CP,WR,WI,C
      1WR,CWI,SC,JCF,RES,D1,D2,DDD,EPS,ITF1,ITFX)
350    IF (IOL.NE.3) GO TO 360
       IF (NG.EQ.0) RETURN
       GO TO 600
360    CONTINUE
       IF (IR.EQ.1.OR.IR.EQ.3) GO TO 500
C---CALCULATION OF CONTROL GAINS:FORMATION OF CONTROL HAMILTONIAN-------
C
C    ---                    ---          ***F AND FT ARE THE OPEN LOOP
C                                           DYNAMICS MATRIX AND TRANSPOSE
C        F       -GM*BI*GMI             ***BI IS NCXNC CONTROL WEIGHTING
C                                           MATRIX
C                                       ***A IS THE NSXNS STATE WEIGHTING
C                                           MATRIX
C        -A           -FT
C    ---                    ---          ***GM IS THE NSXNC CONTROL
C                                             DISTRIBUTION MATRIX
C------------------------------------------------------------------------
       DO 370 I=1,NC
       DO 370 J=1,MH
370    PRO(I,J)=G(J,I)/B(I,I)
       DO 380 I=1,MH
       DO 380 J=1,MH
```

```
110   DDD=DDD+AA(I,K)*WNORMI(K,J)
      DSTORE(I,J)=DDD
120   BA(I,J)=BA(I,J)+DDD
130   CONTINUE
      IF (NO.EQ.0) GO TO 145
      IF (ISET.EQ.1) GO TO 135
      CALL READH (NO,NS,ISAH,HO)
135   CONTINUE
      WRITE (6,1440)
      DO 140 I=1,NO
140   WRITE (6,1390) (HO(I,J),J=1,NS)
145   IF (IM.NE.1) GO TO 150
      CALL MODE (WNORM,HO,CM,NS,NO,NS,2)
150   CONTINUE
      IF (IFDFW.EQ.0) GO TO 170
      IF (ISET.EQ.1) GO TO 155
      CALL READD (NO,NC,ISAL,D)
155   WRITE (6,1470)
      DO 160 I=1,NO
160   WRITE (6,1390) (D(I,J),J=1,NC)
170   CONTINUE
      NOB=0
      IF (NC.EQ.0) GO TO 590
      IF (IOL.EQ.3) GO TO 270
      IF (IR.NE.1.AND.IR.NE.3) GO TO 210
      IF (ISET.EQ.1) GO TO 180
      CALL READG (NS,NC,ISAG,G)
180   CONTINUE
      CALL READFB (NC,NS,FBGC)
      WRITE (6,1400)
      DO 190 I=1,NS
190   WRITE (6,1390) (G(I,J),J=1,NC)
      IF (IM.NE.1) GO TO 200
      CALL MODE (WNORMI,G,BM,NS,NS,NC,0)
200   CONTINUE
      GO TO 330
210   DO 220 I=1,NS
      DO 220 J=1,NS
220   RM(I+MH,J)=0.0
      CALL READAY (NO,ISAA,AY)
      DO 240 I=1,NO
      DO 240 J=1,NS
      DDD=0.D0
      DO 230 K=1,NO
230   DDD=DDD+AY(I,K)*HO(K,J)
240   AA(I,J)=DDD
      WRITE (6,1460)
      DO 250 I=1,NO
```

```
      IF (IDSTAB.EQ.0) GO TC 50
      WRITE (6,1480)
      WRITE (6,1390) (DESTAE(I),I=1,NS)
50    CONTINUE
C---------------EIGENSYSTEM OF THE OPEN LOOP DYNAMICS------------------
      IF (IOL.EQ.0.AND.IQ.EC.0) GO TO 90
      IF (IOL.EQ.0.AND.NC.NE.0) GO TO 90
      DO 60 I=1,NS
      DO 60 J=1,NS
60    GN(I,J)=BA(I,J)
      CALL BALANC (NS,NS,GN,LOW,IHIGH,D1)
      CALL ORTHES (NS,NS,LOW,IHIGH,GN,D2)
      CALL ORTRAN (NS,NS,LOW,IHIGH,GN,D2,SC)
      CALL HQR2 (NS,NS,LOW,IHIGH,GN,CWR,CWI,SC,IERR)
      IF (IERR.NE.0) CALL EREXIT (NS,GN,IERR)
      CALL BALBAK (NS,NS,LOW,IHIGH,D1,NS,SC)
C-----------NORMALIZE AND PRINT OPEN LOOP EIGENSYSTEM------------------
      IWRITE=1
      CALL CNORM (CWR,CWI,SC,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNORMI,HO,CM,
     1NO,NS)
C-----------------------------------------------OPTGROL/OPTGRNO DATA---
      DO 64 I = 1,NS
         WRITE (10,2000) CWR(I), CWI(I)
         WRITE (1,2000) CWR(I), CWI(I)
64    CONTINUE
C---------------------------------------------------------------------
      IF (IOL.EQ.2) RETURN
      IF (IQ.EQ.0.OR.(NC.NE.0.OR.IDSTAB.GT.0)) GO TO 90
      DO 70 I=1,NS
      IF (CWR(I).LT.0.) GO TO 70
      WRITE (5,1490)
      RETURN
70    CONTINUE
      IF (IOL.EQ.3) GO TO 130
      DO 80 I=1,NS
      DO 80 J=1,NS
80    W11(I,J)=SC(I,J)
      CALL MINV (NSQ,W11,NS,DDD,D1,D2)
90    CONTINUE
      IF (IDSTAB.EQ.0) GO TC 130
C------------------- FORM U * DIAG(DESTAB) * U-INV---------------------
      DO 100 J=1,NS
      DO 100 I=1,NS
100   AA(I,J)=WNORM(I,J)*DESTAB(J)
      DO 120 I=1,NS
      DO 120 J=1,NS
      DDD=0.DO
      DO 110 K=1,NS
```

```
     2RMI,DESTAB,AA,BM,CM,JCF,RES,AY,BB,CC,CP,GW,GV,HY,HU,DSTORE,ISAF,IS
     3AH,ISAG,ISAD,IGAM,IRET,PRTT,NROW,NCOL,IRDMAT,ISAA,ISAB)
C=====================================================================
      IMPLICIT REAL*8(A-H,O-Z)
C---------------------------------------------------------------------
      DIMENSION ACL(NS,NS),B(NC,NC),BA(NS,NS),CI(NS),CR(NS),CQ(NS,NS),CW
     1I(NS),CWR(NS),FBGC(NC,NS),FBGE(NS,NO),G(NS,NS),GM(NS,NS),PRO(NS,NS
     2),RC(NO,NO),SC(NS,NS),WR(N2),WI(N2),W11(NS,NS),W21(NS,NS),X(N2,N2)
     3,GN(NS,NS),HO(NO,NS),D1(N2),D2(N2),RM(N2,N2),Q(NG,NG),D(NO,NC),GAM
     4(NS,NG),WNORM(NS,NS),WNORMI(NS,NS),DESTAB(NS),AA(NS,NS),BM(NS,NC),
     5CM(NO,NS),JCF(N2),RES(N2),AY(NO,NO),BB(N2),CC(N2),CP(NS),GW(N2,NG)
     6,GV(N2,NO),HY(NO,N2),HU(NC,N2),DSTORE(NS,NS),PRTT(16,16)
C---------------------------------------------------------------------
      COMMON /PROG/ IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,IDEB
     1UG,ISET,IREG,IPSD,IYU,INORM
C---------------------------------------------------------------------
      REAL*4 FMT(20)
C-----------------OUTPUT OPTIONS--------------------------------------
C---IOL=1 IF THE OPEN LOOP EIGENSYSTEM IS DESIRED--OTHERWISE IOL=0
C---IQ=1 IF THE RMS VALUES OF THE CONTROL AND STATE ARE TO BE FOUND
C---IR=0 IF OPTIMAL FILTER AND REGULATOR EIGENSYSTEMS ARE TO BE FOUND
C---IR=1 IF EXTERNAL C MATRIX IS SUPPLIED
C---IR=2 IF EXTERNAL K IS SUPPLIED
C---IR=3 IF EXTERNAL C AND K ARE SUPPLIED
C---ISS=1 IF STEADY STATE VALUES ARE TO BE DETERMINED
C---IM=1 IF MODAL STATES DESIRED
C---------------------------------------------------------------------
      NSQ=NS*NS
      MH=NS
      M=N2
      CALL CHECK (EPS,NC,NG,NO,IRET)
      IF (IRET.EQ.1) RETURN
      IF (ISET.EQ.1) GO TO 20
      CALL RDMAT(BA,G,HO,D,GAM,FBGC,FBGE,AY,B,NS,NC,NO,NG,IRDMAT,IFDFW)
      CALL READF (NS,ISAF,BA)
      IF (IDSTAB.EQ.0) GO TC 10
      WRITE (5,1800)
      CALL RDREAL (ANSR)
      DSTAB=ANSR
      DO 10 I=1,NS
      DESTAB(I)=DSTAB
10    CONTINUE
      GO TO 30
20    CALL SETUP (BA,G,GAM,HO,D,NS,NC,NG,NO)
30    CONTINUE
      WRITE (6,1380)
      DO 40 I=1,NS
40    WRITE (6,1390) (BA(I,J),J=1,NS)
```

```
C-----------------------------------------------------------------------
      END
C=======================================================================
      SUBROUTINE RAPRNT (NMAX,M,N,L,A,IDIM,FMT)
      REAL*8 A(NMAX,N)
      DIMENSION FMT(IDIM)
      NU=L
      DO 20 NL=1,N,L
      IF (NU.GT.N) NU=N
      DO 10 I=1,M
10    WRITE (6,FMT) (A(I,J),J=NL,NU)
      WRITE (6,30)
20    NU=NU+L
      RETURN
30    FORMAT (1X)
      END
C=======================================================================
      SUBROUTINE RGAIN (M,NS,NC,NOB,WR,WI,VF,GN,W11,TCB,W21,LT,C,CI,CT,M
     1HS,MT)
      IMPLICIT REAL*8 (A-H,C-Z)
      DIMENSION WR(M),WI(M),VF(M,M),GN(NS,NS)
      DIMENSION W11(NS,NS),TCB(M,M),W21(NS,NS),LT(NS),MT(NS)
      DIMENSION C(NS),CI(NS),CT(NS,NS)
      K=1
      KP=1
      KN=1
      NRZEV=0
      NCPZEV=0
10    IF (K.GT.M) GO TO 210
C-----------------------------------------------------------------------
C CHECK FOR EIGVAL AT OR NEAR J-OMEGA AXIS TO INCLUDE IN E-L EIGSYS
C  TURN FIRST ONE POSITIVE AND SECOND ONE NEGATIVE
C-----------------------------------------------------------------------
      EIGVR=DABS(WR(K))
      IF (EIGVR.GE.1.D-10) GO TO 60
      IF (WI(K)) 40,20,40
20    NRZEV=NRZEV+1
      IF (NRZEV.GT.1) GO TO 30
      WR(K)=EIGVR
      GO TO 80
30    WR(K)=-EIGVR
      WRITE (6,290)
      GO TO 150
40    NCPZEV=NCPZEV+1
      IF (NCPZEV.GT.1) GO TO 50
      WR(K)=EIGVR
      WR(K+1)=EIGVR
      GO TO 110
```

```
50    WR(K)=-EIGVR
      WR(K+1)=-EIGVR
      WRITE (6,300)
      GO TO 180
60    IF (WR(K)) 140,70,70
70    IF (WI(K)) 110,80,110
C---------------EIGENVECTOR FOR REAL EIGENVALUE,POSITIVE--------------
80    IF (NOB.EQ.0) GO TO 100
      DO 90 J=1,M
90    TCB(J,KP)=VF(J,K)
100   KP=KP+1
      K=K+1
      GO TO 10
C----------EIGENVECTOR FOR COMPLEX EIGENVALUE,POSITIVE REAL PART-------
110   IF (NOB.EQ.0) GO TO 130
      DO 120 J=1,M
      FR=VF(J,K)
      FI=-VF(J,K+1)
      TCB(J,KP)=FR+FI
120   TCB(J,KP+1)=FR-FI
130   KP=KP+2
      K=K+2
      GO TO 10
140   IF (WI(K)) 180,150,180
C-----------EIGENVECTOR FOR REAL EIGENVALUE,NEGATIVE REAL PART---------
150   C(KN)=WR(K)
      CI(KN)=WI(K)
      IF (NOB.NE.0) GO TO 170
      KNS=KN+NS
      DO 160 J=1,M
160   TCB(J,KNS)=VF(J,K)
170   KN=KN+1
      K=K+1
      GO TO 10
C-----------EIGENVECTOR FOR COMPLEX EIGENVALUE,NEGATIVE REAL PART------
180   RR=WR(K)
      RI=WI(K)
      C(KN)=RR
      C(KN+1)=RR
      CI(KN)=RI
      CI(KN+1)=-RI
      IF (NOB.NE.0) GO TO 200
      KNS=KN+NS
      DO 190 J=1,M
      FR=VF(J,K)
      FI=-VF(J,K+1)
      TCB(J,KNS)=FR+FI
190   TCB(J,KNS+1)=FR-FI
```

```
200   KN=KN+2
      K=K+2
      GO TO 10
210   CONTINUE
      IF (NOB.NE.0) GO TO 240
C------------------------FORMATION OF W11----------------------------
      DO 220 I=1,NS
      DO 220 J=1,NS
      W11(I,J)=TCB(I,J+NS)
220   CT(I,J)=W11(I,J)
C------------------------FORMATION OF W21----------------------------
      DO 230 I=1,NS
      DO 230 J=1,NS
230   W21(I,J)=TCB(I+NS,J+NS)
240   IF (NOB.EQ.0) GO TO 260
      DO 250 I=1,NS
      DO 250 J=1,NS
      W21(I,J)=-TCB(I,J)
250   W11(I,J)=TCB(I+NS,J)
260   CONTINUE
C-------------------------INVERT W11---------------------------------
      NSQ=NS*NS
      CALL MINV (NSQ,W11,NS,DETC,LT,MT)
C------------------CALCULATE THE RGAIN MATRIX------------------------
      DO 270 IL=1,NS
      DO 270 JL=1,NS
      GN(IL,JL)=0.D0
      DO 270 KL=1,NS
270   GN(IL,JL)=GN(IL,JL)+W21(IL,KL)*W11(KL,JL)
      IF (NOB.EQ.0) RETURN
      DO 280 I=1,NS
      DO 280 J=1,NS
280   CT(I,J)=W11(J,I)
      RETURN
C--------------------------------------------------------------------
290   FORMAT (1X,51H EULER-LAGRANGE EQUATIONS HAVE A REAL EIGENVALUE AT,
     114H OR NEAR ZERO./)
300   FORMAT (1X,49H EULER-LAGRANGE EQUATIONS HAVE A COMPLEX PAIR OF ,40
     1HEIGENVALUES AT OR NEAR THE J-OMEGA AXIS.)
      END
C====================================================================
      SUBROUTINE MINV (NSQ,A,N,D,L,M)
      IMPLICIT REAL*8 (A-H,C-Z)
      DIMENSION A(NSQ),L(N),M(N)
      DOUBLE PRECISION A,D,EIGA,HOLD
      NM=N*N
      D=1.0D0
      NK=-N
```

```
      DO 180 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
      IF (DABS(BIGA)-DABS(A(IJ))) 10,20,20
10    BIGA=A(IJ)
      L(K)=I
      M(K)=J
20    CONTINUE
C---------------------------INTERCHANGE ROWS---------------------------
      J=L(K)
      IF (J-K) 50,50,30
30    KI=K-N
      DO 40 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
40    A(JI)=HOLD
C---------------------------INTERCHANGE COLUMNS------------------------
50    I=M(K)
      IF (I-K) 80,80,60
60    JP=N*(I-1)
      DO 70 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
70    A(JI)=HOLD
C-----------------------DIVIDE COLUMN BY MINUS PIVOT-------------------
C------------------(VALUE OF PIVOT ELEMENT IS CONTAINED IN BIGA)--------
80    IF (BIGA) 100,90,100
90    D=0.0D0
      RETURN
100   DO 120 I=1,N
      IF (I-K) 110,120,110
110   IK=NK+I
      A(IK)=A(IK)/(-BIGA)
120   CONTINUE
C---------------------------REDUCE MATRIX------------------------------
      DO 150 I=1,N
      IK=NK+I
```

```
      HOLD=A(IK)
      IJ=I-N
      DO 150 J=1,N
      IJ=IJ+N
      IF (I-K) 130,150,130
130   IF (J-K) 140,150,140
140   KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
150   CONTINUE
C-------------------------DIVIDE ROW BY PIVOT-----------------------
      KJ=K-N
      DO 170 J=1,N
      KJ=KJ+N
      IF (J-K) 160,170,160
160   A(KJ)=A(KJ)/BIGA
170   CONTINUE
C-------------------------PRODUCT OF PIVOTS-------------------------
      D=D*BIGA
C-----------------------REPLACE PIVOT BY RECIPROCAL-----------------
      A(KK)=(1.0D0)/BIGA
180   CONTINUE
C--------------------FINAL ROW AND COLUMN INTERCHANGE---------------
      K=N
190   K=(K-1)
      IF (K) 260,260,200
200   I=L(K)
      IF (I-K) 230,230,210
210   JQ=N*(K-1)
      JR=N*(I-1)
      DO 220 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
220   A(JI)=HOLD
230   J=M(K)
      IF (J-K) 190,190,240
240   KI=K-N
      DO 250 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
250   A(JI)=HOLD
      GO TO 190
260   K=0
      RETURN
      END
```

```
C=====================================================================
      SUBROUTINE SCOV (NL,WI,WLI,VL1,VL2,NR,WR,WRI,VR1,VR2,Q,X)
      REAL*8 VL1(NL),VL2(NL),WL(NL,NL),WLI(NL,NL),X(NL,NR),Q(NL,NR),
     1   VR1(NR),VR2(NR),WR(NR,NR),WRI(NR,NR)
      REAL*8 A,B,C,D,K1,K2,K3,K4
10    DO 20 I=1,NL
      DO 20 J=1,NR
      X(I,J)=0.
      DO 20 II=1,NL
20    X(I,J)=X(I,J)+WLI(I,II)*Q(II,J)
      DO 40 I=1,NL
      DO 40 J=1,NR
      Q(I,J)=0.
      DO 30 JJ=1,NR
30    Q(I,J)=Q(I,J)+X(I,JJ)*WRI(J,JJ)
40    CONTINUE
      I=1
50    IF (VL2(I)) 60,110,60
60    J=1
70    IF (VR2(J)) 80,90,80
80    A=VL1(I)+VR1(J)
      B=-2.*VL2(I)*VR2(J)
      C=A**2+VL2(I)**2+VR2(J)**2
      D=C**2-B**2
      K1=A*C/D
      K2=-(VR2(J)*C+VL2(I)*B)/D
      K3=-(VR2(J)*B+VL2(I)*C)/D
      K4=-A*B/D
      I1=I+1
      J1=J+1
      X(I,J)=+K1*Q(I,J)+K2*C(I,J1)+K3*Q(I1,J)+K4*Q(I1,J1)
      X(I,J1)=-K2*Q(I,J)+K1*Q(I,J1)-K4*Q(I1,J)+K3*Q(I1,J1)
      X(I1,J)=-K3*Q(I,J)-K4*Q(I,J1)+K1*Q(I1,J)+K2*Q(I1,J1)
      X(I1,J1)=+K4*Q(I,J)-K3*Q(I,J1)-K2*Q(I1,J)+K1*Q(I1,J1)
      J=J+2
      GO TO 100
90    A=VR1(J)+VI1(I)
      B=A**2+VL2(I)**2
      K1=A/B
      K2=VL2(I)/B
      X(I,J)=K1*Q(I,J)-K2*Q(I+1,J)
      X(I+1,J)=K2*Q(I,J)+K1*Q(I+1,J)
      J=J+1
100   IF (J.LE.NR) GO TO 70
      I=I+2
      GO TO 160
110   J=1
120   IF (VR2(J)) 130,140,130
```

```
130   A=VR1(J)+VL1(I)
      B=A**2+VR2(J)**2
      K1=A/B
      K2=VR2(J)/B
      X(I,J)=K1*Q(I,J)-K2*Q(I,J+1)
      X(I,J+1)=K2*Q(I,J)+K1*Q(I,J+1)
      J=J+2
      GO TO 150
140   X(I,J)=Q(I,J)/(VR1(J)+VL1(I))
      J=J+1
150   IF (J.LE.NR) GO TO 120
      I=I+1
160   IF (I.LE.NL) GO TO 50
      DO 170 I=1,NL
      DO 170 J=1,NR
      Q(I,J)=0.
      DO 170 II=1,NL
170   Q(I,J)=Q(I,J)+WL(I,II)*X(II,J)
      DO 190 I=1,NL
      DO 190 J=1,NR
      X(I,J)=0.
      DO 180 JJ=1,NR
180   X(I,J)=X(I,J)+Q(I,JJ)*WR(J,JJ)
190   CONTINUE
      RETURN
      END
C===================================================================
      SUBROUTINE MODE (WNORM,G,GNORM,NS,N1,N2,ICON)
C                                                                  =
C  WNORM  TRANSFORMATION MATRIX U OR U-INV                         =
C  NS     NO. OF STATE                                             =
C  NC     NO.  OF INPUTS OR OUTPUTS                                =
C  ICON  CONTROL FLAG TO INDICATE WHICH TRANSFORMATION             =
C         0 =  MODAL G                                             =
C         1 =  MODAL GAMMA                                         =
C         2 =  MODAL H                                             =
C         3 =  MODAL C                                             =
C         4 =  MODAL K                                             =
C         5 =  CONTROL EIGENVECTOR MATRIX                          =
C         6 =  MEASUREMENT EIGENVECTOR MATRIX                      =
C===================================================================
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION WNORM(NS,NS),G(N1,N2),GNORM(N1,N2)
      DO 10 I=1,N1
      DO 10 J=1,N2
10    GNORM(I,J)=0.
      IPOINT=ICON+1
      GO TO (20,20,90,90,20,90,90), IPOINT
```

```
20    DO 30 J=1,N2
      DO 30 I=1,NS
      DO 30 K=1,NS
30    GNORM(I,J)=GNORM(I,J)+WNORM(I,K)*G(K,J)
      GO TO (40,70,90,90,80), IPOINT
40    WRITE (6,170)
50    DO 60 I=1,NS
60    CONTINUE
C0    WRITE (6,230) (GNORM(I,J),J=1,N2)
      RETURN
70    WRITE (6,180)
      GO TO 50
80    WRITE (6,240)
      GO TO 50
90    DO 100 J=1,NS
      DO 100 I=1,N1
      DO 100 K=1,NS
100   GNORM(I,J)=GNORM(I,J)+G(I,K)*WNORM(K,J)
      GO TO (110,110,110,12C,110,130,140), IPOINT
110   WRITE (6,190)
      GO TO 150
120   WRITE (6,200)
      GO TO 150
130   WRITE (6,210)
      GO TO 150
140   WRITE (6,220)
150   DO 160 I=1,N1
160   WRITE (6,230) (GNORM(I,J),J=1,NS)
      RETURN
C-----------------------------------------------------------------------
170   FORMAT (//,5X,45HMODAI CONTROL DISTRIBUTION MATRIX......TI*G..,//)
180   FORMAT (//,5X,50HMODAI PROCESS NOISE DISTRIBUTION MATRIX...TI*GAM.
     1.,//)
190   FORMAT (//,5X,45HMODAI MEASUREMENT SCALING MATRIX...H(BAR)*T..,//)
200   FORMAT (//,5X,45HTHE MODAL CONTROL GAINS................C*T..,//)
210   FORMAT (//,5X,45HCONTROL EIGENVECTOR MATRIX..............C*M..,//)
220   FORMAT (//,5X,45HMEASUREMENT EIGENVECTOR MATRIX.....H(BAR)*M..,//)
230   FORMAT (1X,(2X,1P6D14.6))
240   FORMAT (//,5X,45HMODAI FILTER STEADY STATE GAINS........TI*K..,//)
      END
C=======================================================================
      SUBROUTINE CNORM (WZ,WY,VEC,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNORMI,H
     1O,CM,N1,N2)
C                                                                      =
C         WZ(I)      REAL PART OF I-TH EIGENVALUE                      =
C                                                                      =
C         WY(I)      COMPLEX PART OF I-TH EIGENVALUE                   =
C                                                                      =
```

```
C          VEC        MATRIX CF RIGHT EIGENVECTORS STORED IN REAL FORM   =
C                      FROM HCR2                                         =
C          NS         NO. OF STATES                                      =
C                                                                        =
C          IWRITE     FLAG TO CONTROL FORMATS FOR DIFFERENT EIGHENSYSTEMS=
C                                                                        =
C          WNORM      NORMALIZED MATRIX U OF RIGHT EIGENVECTORS STORED   =
C                      BY COLUMNS IN REAL FORM                           =
C          WNORMI     U-INVERSE 2*CONGUGATE OF LEFT EIGENVECTORS         =
C                       STORED BY ROW IN REAL FORM                       =
C          NSQ,DDD,D1,D2 - ARGUMENTS PASSED TO MINV                      =
C=======================================================================
      IMPLICIT REAL*8 (A-H,C-Z)
      REAL*8 FIELD,COMMA,SEMCOL,RIGHT,FMT
      DIMENSION WZ(NS),WY(NS),VEC(NS,NS),WNORM(NS,NS),WNORMI(NS,NS),STOR
     1E(6),D1(NS),D2(NS),FMT(14),HO(N1,N2),CM(N1,N2)
      DATA FIELD/5HE12.5/,COMMA/5H,',',/,SEMCOL/5H,':',/,RIGHT/1H)/,FMT/
     16H(1X,1P,13*1H /,SEMEND/4H,':'/
C----------NORMALIZE COMPLEX EIGENVECTORS BY LARGEST ELEMENT-----------
      KK=0
      LR=0
      LC=0
      DO 50 K=1,NS
      IF (KK.EQ.1) GO TO 40
      IF (DABS(WY(K)).LT.1.D-10) GO TO 50
      LC=LC+1
      EMAX=0.D0
      DO 20 I=1,NS
      CMOD=VEC(I,K)**2+VEC(I,K+1)**2
      IF (CMOD-EMAX) 20,10,10
10    EMAX=CMOD
      M=I
20    CONTINUE
      VMR=VEC(M,K)
      VMI=VEC(M,K+1)
      DO 30 I=1,NS
      VR=VEC(I,K)
      VI=VEC(I,K+1)
      VECRN=(VR*VMR+VI*VMI)/EMAX
      VECIN=(-VR*VMI+VI*VMR)/EMAX
      WNORM(I,K)=VECRN
      WNORM(I,K+1)=VECIN
30    CONTINUE
      KK=1
      GO TO 50
40    KK=0
50    CONTINUE
C-----------NORMALIZE REAL EIGENVECTORS BY THE TOTAL LENGTH-----------
```

```
      DO 80 K=1,NS
      IF (DABS(WY(K)).GE.1.D-10) GO TO 80
      LR=LR+1
      REMOD=0.D0
      DO 60 I=1,NS
60    REMOD=VEC(I,K)**2+REMCD
      RMOD=DSQRT(REMOD)
      DO 70 I=1,NS
      RVEC=VEC(I,K)/RMOD
      WNORM(I,K)=RVEC
70    CONTINUE
80    CONTINUE
      GO TO (90 100,110,120,130), IWRITE
90    WRITE (6,320)
      GO TO 140
100   WRITE (6,330)
      GO TO 140
110   WRITE (6,340)
      GO TO 140
120   WRITE (6,350)
      GO TO 140
130   WRITE (6,360)
140   KK=0
      NPRTW=0
      NFMTW=1
      DO 180 I=1,NS
      IF (KK.EQ.1) GO TO 170
      IF (DABS(WY(I)).GT.1.D-10) KK=1
C----PRINT OUT NO MORE THAN 6 WORDS, NOT SEPARATING COMPLEX EIGVAL------
      IF (NPRTW.LT.5.OR.(NPRTW.EQ.5.AND.KK.EQ.0)) GO TO 150
      FMT(NFMTW+1)=RIGHT
      WRITE (6,FMT) (STORE(J),J=1,NPRTW)
      NPRTW=0
      NFMTW=1
150   NPRTW=NPRTW+1
      NFMTW=NFMTW+1
      IF (KK.EQ.1) GO TO 160
      STORE(NPRTW)=WZ(I)
      FMT(NFMTW)=FIELD
      NFMTW=NFMTW+1
      FMT(NFMTW)=SEMCOL
      GO TO 180
160   STORE(NPRTW)=WZ(I)
      FMT(NFMTW)=FIELD
      FMT(NFMTW+1)=COMMA
      STORE(NPRTW+1)=WY(I)
      FMT(NFMTW+2)=FIELD
      FMT(NFMTW+3)=SEMCOL
```

```
      NFMTW=NFMTW+3
      NPRTW=NPRTW+1
      GO TO 180
170   KK=0
180   CONTINUE
      FMT(NFMTW)=SEMEND
      FMT(NFMTW+1)=RIGHT
      WRITE (6,FMT) (STORE(J),J=1,NPRTW)
      IF (IWRITE.NE.1) GO TC 190
      WRITE (6,370)
      GO TO 200
190   WRITE (6,380)
200   CALL RAPRNT (NS,NS,NS,6,WNORM,4,'(6(1X,1PD13.6))')
      GO TO (230,210,210,22C,220), IWRITE
210   CALL MODE (WNORM,HO,CM,NS,N1,N2,5)
      GO TO 230
220   CALL MODE (WNORM,HO,CM,NS,N1,N2,6)
230   GO TO (240,250,260,27C,280), IWRITE
240   WRITE (6,390)
      GO TO 290
250   WRITE (6,400)
      GO TO 290
260   WRITE (6,410)
      GO TO 290
270   WRITE (6,420)
      GO TO 290
280   WRITE (6,430)
C-----------------SAVE U-INVERSE OPEN LOOP IN WNORMI-----------------
290   IF (IWRITE .GT. 1) GO TO 310
      DO 300 I=1,NS
      DO 300 J=1,NS
300   WNORMI(I,J)=WNORM(I,J)
      CALL MINV (NSQ,WNORMI,NS,DDD,D1,D2)
      CALL RAPRNT (NS,NS,NS,6,WNORMI,4,'(6(1X,1PD13.6))')
      RETURN
310   CALL MINV (NSQ,WNORM,NS,DDD,D1,D2)
      CALL RAPRNT (NS,NS,NS,6,WNORM,4,'(6(1X,1PD13.6))')
      RETURN
C--------------------------------------------------------------------
320   FORMAT (//5X,42HOPEN LOOP EIGENVALUES..........DET(SI-F)...//)
330   FORMAT (//5X,46HC-LOOP OPTIMAL REG. E-VALUES...DET(SI-F+G*C)...//)
340   FORMAT (//5X,46HC-LOOP SUBOPT. REG. E-VALUES...DET(SI-F+G*C)...//)
350   FORMAT (//5X,46HC-LOOP OPTIMAL EST. E-VALUES...DET(SI-F+K*H)...//)
360   FORMAT (//5X,46HC-LOOP SUBOPT. EST. E-VALUES...DET(SI-F+K*H)...//)
370   FORMAT (//5X,46HOPEN LOOP RIGHT EIGENVECTOR MATRIX.......T....,//)
380   FORMAT (//5X,46HC-LOOP RIGHT EIGENVECTOR MATRIX..........M....,//)
390   FORMAT (//5X,46HOPEN LOOP LEFT EIGENVECTOR MATRIX......T-INV..,//)
400   FORMAT (//5X,46HC-LOOP OPT. REG. LEFT E-VECTOR MATRIX..M-INV..,//)
```

```
410   FORMAT (//5X,46HC-LOOP SUBOPT-REG. LEFT E-VECTOR MATRIX..M-INV.//)
420   FORMAT (//5X,46HC-LOOP OPT. FILTER LEFT E-VECTOR MATRIX..M-INV.//)
430   FORMAT (//5X,51HC-LOOP SUBOPT. FILTER LEFT E-VECTOR MATRIX..M-INV.
     1.//)
      END
C=====================================================================
      SUBROUTINE TF (N,NM,NSQ,A,AA,M,B,BM,L,C,CM,IFDFW,D,BB,CC,CP,
     1     EVR,EVI,PR,PI,SC,JCF,RES,D1,D2,DDD,EPS,ITF,ITFX)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(N,N),AA(N,N),B(N,M),BM(N,M),C(L,N),CM(L,N),D(L,M),BB(N
     1),CC(N),CP(N),EVR(N),EVI(N),PR(N),PI(N),SC(N,N),JCF(N),RES(N),D1(N
     2),D2(N)
C--SAVE COMPUTATION ON OL AND CL SYS WITH MODAL WORK DONE IN OPTSYS-----
      IF (ITFX .EQ. 1) GO TO 50
      IF (ITFX .EQ. 2) GO TO 10
      CALL POLES (N,NM,A,AA,M,B,L,C,PR,PI,D1,D2,JCF,SC)
C--------------COMPUTE MODAL MATRICES FOR RESIDUES--------------------
10    DO 20 I=1,N
      DO 20 J=1,N
20    AA(I,J)=SC(I,J)
      DO 30 I=1,L
      DO 30 J=1,N
      CM(I,J)=0.D0
      DO 30 K=1,N
30    CM(I,J)=CM(I,J)+C(I,K)*AA(K,J)
      CALL MINV (NSQ,AA,N,DDD,D1,D2)
      DO 40 I=1,N
      DO 40 J=1,M
      BM(I,J)=0.D0
      DO 40 K=1,N
40    BM(I,J)=BM(I,J)+AA(I,K)*B(K,J)
50    CONTINUE
      DO 60 I=1,M
      DO 60 J=1,L
      IF (ITF .NE. 3) CALL ZEROS (I,J,IFDFW,N,NM,A,AA,M,B,L,C,D,BB,CC,CP
     1,EVR,EVI,D1,D2,EPS,ITFX)
      IF (ITF .NE. 2) CALL RESID (I,J,N,JCF,M,BM,L,CM,PR,PI,RES,BB,CC,1)
60    CONTINUE
      RETURN
      END
C=====================================================================
      SUBROUTINE POLES (N,NM,A,AA,M,B,L,C,EVR,EVI,D1,D2,JCF,SC)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(N,N),AA(N,N),B(N,M),C(L,N),EVR(N),EVI(N),D1(N),D2(N),J
     1CF(N),SC(N,N)
      DO 10 I=1,N
      DO 10 J=1,N
10    AA(I,J)=A(I,J)
```

```
      CALL ORTHES (NM,N,LOW,IHIGH,AA,D2)
      CALL ORTRAN (NM,N,LOW,IHIGH,AA,D2,SC)
      CALL HQR2 (NM,N,LOW,IHIGH,AA,EVR,EVI,SC,IERR)
      IF (IERR .NE. 0) GO TC 30
      CALL BALBAK (NM,N,LOW,IHIGH,D1,N,SC)
      WRITE (6,40)
      DO 20 I=1,N
C--------------------------------------------------OPTGRCM DATA-----------
      WRITE (4,2000) EVR(I),EVI(I)
C-------------------------------------------------------------------------
20    WRITE (6,50) EVR(I),EVI(I)
      RETURN
30    WRITE (5,60)
      RETURN
C-------------------------------------------------------------------------
40    FORMAT (///,28H TF DENOMINATOR EIGENVALUES:,/)
50    FORMAT (/,2X,3H  (,F13.6,4H)+J(,F13.6,1H))
60    FORMAT (35H FAILURE IN HQR2, CALCULATING POLES)
C--------------------------------------------------OPTGRCM DATA-----------
2000  FORMAT(5X,2D30.14)
C-------------------------------------------------------------------------
      END
C=========================================================================
      SUBROUTINE ZEROS (K1,K2,IFDFW,N,NM,A,AA,M,B,L,C,D,BB,CC,CP,EVR,EVI
     1,     D1,D2,EPS,ITFX)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(N,N),AA(N,N),B(N,M),C(L,N),D(L,M),BB(N),CC(N),CP(N),EV
     1R(N),EVI(N),D1(N),D2(N)
      DOUBLE PRECISION SCL,DABS
C------------------------------------------OPGRAPH DATA -------------------
      IF(ITFX.EQ.1) IFL = 10
      IF(ITFX.EQ.2) IFL = 1
      IF(ITFX.EQ.3) IFL = 4
C-------------------------------------------------------------------------
      DO 10 I=1,N
      BB(I)=B(I,K1)
      CC(I)=C(K2,I)
      DO 10 J=1,N
10    AA(I,J)=A(I,J)
      WRITE (6,90) K1,K2
      IF (IFDFW .EQ. 0) GO TO 20
      H=D(K2,K1)
      IF (DABS(H).LE.EPS) GC TO 20
      JJ=N
      GO TO 50
20    NN=N-1
      DO 30 I=1,NN
```

```
      CALL CCOMP (N,NM,AA,CC,CP)
      IF (DABS(H).GT.EPS) GO TO 40
30    CONTINUE
      H=SCL(N,BB,CC)
      WRITE (6,100) H
C-----------------------------------OPGRAPH DATA FOR "0" ORDER-------
          KK=ITFX
          WRITE (IFL,300) KK,K2,K1
          ORDER = 0.0
          WRITE (IFL,301) ORDER,H
C-------------------------------------------------------------------
      GO TO 70
40    JJ=N-I
50    WRITE (6,110) JJ,H
C-----------------------------------OPGRAPH DATA -------------------
          KK=ITFX
          WRITE (IFL,300) KK,K2,K1
          ORDER = FLOAT(JJ)
          WRITE (IFL,301) ORLER,H
C-------------------------------------------------------------------
      CALL ACOMP (N,NM,AA,BE,CC,H)
      CALL BALANC (NM,N,AA,IOW,IHIGH,D1)
      CALL ORTHES (NM,N,LOW,IHIGH,AA,D2)
      CALL HQR (NM,N,LOW,IHIGH,AA,EVR,EVI,IERR)
      IF (IERR .NE. 0) GO TO 80
      WRITE (6,120)
      DO 60 I=1,N
60    WRITE (6,130) EVR(I),EVI(I)
C-----------------------------------OPGRAPH DATA -------------------
          WRITE (IFL,302) K2,K1
      DO 63 LL = 1,N
63        WRITE (IFL,301) EVR(LL),EVI(LL)
C-------------------------------------------------------------------
70    RETURN
80    WRITE (5,140)
      RETURN
C-------------------------------------------------------------------
90    FORMAT (///,17H TF FOR INPUT NO.,I3,15H AND OUTPUT NO.,I3,1H:)
100   FORMAT (//,5X,27HNO FINITE ZEROS.  TF GAIN =,E12.4)
110   FORMAT (/,3X,20HORDER OF NUMERATOR =,I3,9X,9HTF GAIN =,E12.4)
120   FORMAT (/,3X,57HNUMERATOR EIGENVALUES (INCLUDING EXTRANEOUS ZERO V
     1ALUES):)
130   FORMAT (/,4X,1H(,F13.6,4H)+J(,F13.6,1H))
140   FORMAT (52H FAILURE IN HQR CALCULATING TRANSFER FUNCTION ZEROES)
C-------------------------------------------------OPGRAPH DATA----------
300   FORMAT(5X,3I5)
301   FORMAT(5X,2D30.14)
```

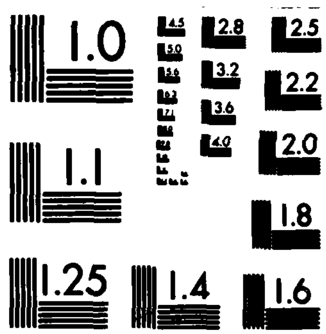
AD-A154 552 DEVELOPMENT OF REDUCED ORDER MODELS FOR CONTROL SYSTEM DESIGN USING THE OPTSYSX PROGRAM(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA S W NELSON DEC 84 2/2

UNCLASSIFIED F/G 9/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
302    FORMAT(5X,2I5)
C-----------------------------------------------------------------------
       END
C=======================================================================
       SUBROUTINE ACOMP (N,NM,A,B,C,H)
       REAL*8 A,B,C,H
       DIMENSION A(NM,N),B(N),C(N)
       DO 10 I=1,N
       DO 10 J=1,N
10     A(I,J)=A(I,J)-B(I)*C(J)/H
       RETURN
       END
C=======================================================================
       SUBROUTINE CCOMP (N,NM,A,C,CC)
       REAL*8 A,C,CC
       DIMENSION A(NM,N),C(N),CC(N)
       DO 10 I=1,N
       CC(I)=0.
       DO 10 J=1,N
10     CC(I)=CC(I)+C(J)*A(J,I)
       DO 20 I=1,N
20     C(I)=CC(I)
       RETURN
       END
C=======================================================================
       FUNCTION SCL (N,B,C)
       REAL*8 B,C,SCL
       DIMENSION B(N),C(N)
       SCL=0.
       DO 10 I=1,N
10     SCL=SCL+C(I)*B(I)
       RETURN
       END
C=======================================================================
       SUBROUTINE RESID (K1,K2,N,JCF,M,BM,L,CM,PR,PI,RES,BB,CC,IPT)
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION JCF(N),BM(N,M),CM(L,N),PR(N),PI(N),RES(N),BB(N),CC(N),PR
      1T(4)
       DATA SN/8H*SIN(B*T/,R1/8H        */,R2/8HEXP(A*T)/,ED/1H)/
       DATA ZERO/0.D0/,T1/4H*T**/,BLANK/8H        /,CS/8H*COS(B*T/
C-------------TEMPORARY MOD TILL JCF IS CALCULATED----------------------
       DO 10 I=1,N
10     JCF(I)=0
C----------------------TEMPORARY MOD-----------------------------------
       IF (IPT .EQ. 1) WRITE (6,170)
       DO 20 I=1,N
       BB(I)=BM(I,K1)
20     CC(I)=CM(K2,I)
```

```
C--------------------LOOP THROUGH THE POLES-----------------------
      I=0
30    I=I+1
      IF (I .GT. N) GO TO 160
      IF (JCF(I) .EQ. 1) GO TO 60
      IF (DABS(PI(I)) .LT. 1.D-10) GO TO 50
C--------COMPUTE SIMPLE COMPLEX POLE RESIDUES AND PRINT BOTH----------
      RES(I)=CC(I)*BB(I)+CC(I+1)*BB(I+1)
      RES(I+1)=CC(I)*BB(I+1)-CC(I+1)*BB(I)
      IF (IPT .EQ. 0) GO TO 40
      PRT(1)=BLANK
      PRT(2)=R2
      IF (PI(I) .EQ. 0.D0) PRT(2)=BLANK
      PRT(3)=CS
      PRT(4)=ED
      WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)
      I=I+1
      PRT(3)=SN
      WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)
      GO TO 30
40    I=I+1
      GO TO 30
50    CONTINUE
C---------------COMPUTE SIMPLE REAL POLE RESIDUE-------------------
      RES(I)=CC(I)*BB(I)
      IF (IPT .EQ. 0) GO TO 30
      PRT(1)=R1
      PRT(2)=R2
      PRT(3)=BLANK
      PRT(4)=BLANK
      WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)
      GO TO 30
C---------LOOK AHEAD TO DETERMINE SIZE OF THE JORDAN BLOCK------------
60    K=1
      KT=N-I
      DO 70 J=I,KT
      IF (JCF(J) .EQ. 0) GO TO 80
70    K=K+1
80    CONTINUE
      IF (DABS(PI(I)) .LT. 1.D-10) GO TO 110
C--------COMPUTE REPEATED COMPLEX POLE AND PRINT OUT ALL FOUR---------
      K=1
      RES(I)=CC(I)*BB(I)+CC(I+1)*BB(I+1)+CC(I+2)*BB(I+2)+CC(I+3)*BB(I+3)
      RES(I+1)=CC(I)*BB(I+1)-CC(I+1)*BB(I)+CC(I+2)*BB(I+3)-CC(I+3)*BB(I      12)
      RES(I+2)=CC(I)*BB(I+3)+CC(I+1)*BB(I+2)
      RES(I+3)=CC(I)*BB(I+3)-CC(I+1)*BB(I+2)
      IF (IPT .EQ. 0) GO TO 100
      PRT(1)=R1
```

```
      PRT(2)=R2
      IF (DABS(PR(I)) .GT. 1.D-10) GO TO 90
      PRT(1)=BLANK
      PRT(2)=BLANK
90    PRT(3)=CS
      PRT(4)=ED
      WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)
      PRT(3)=SN
      I=I+1
      WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)
      PRT(1)=T1
      PRT(2)=R2
      IF (DABS(PR(I)) .LT. 1.D-10) PRT(2)=BLANK
      PRT(3)=CS
      I=I+1
      WRITE (6,190) PR(I),PI(I),RES(I),PRT(1),K,(PRT(J),J=2,4)
      PRT(3)=SN
      I=I+1
      WRITE (6,190) PR(I),PI(I),RES(I),PRT(1),K,(PRT(J),J=2,4)
      GO TO 30
100   I=I+3
      GO TO 30
C----COMPUTE REPEATED REAL POLE RESIDUE AND PRINT OUT ALL K OF THEM-----
110   CONTINUE
      KT=I+K-1
      NN=0
      DO 130 J=I,KT
      NN=NN+1
      RES(J)=ZERO
      DO 120 JJ=J,KT
120   RES(J)=RES(J)+BB(JJ)*CC(JJ-NN+1)
130   CONTINUE
      IF (IPT .EQ. 0) GO TO 150
      NN=0
      PRT(1)=T1
      PRT(2)=R2
      PRT(3)=BLANK
      PRT(4)=BLANK
      DO 140 J=I,KT
      WRITE (6,190) PR(J),PI(J),RES(J),PRT(1),NN,(PRT(JJ),JJ=2,4)
140   NN=NN+1
      GO TO 30
150   I=KT
      GO TO 30
160   CONTINUE
      RETURN
C-----------------------------------------------------------------------
170   FORMAT (//,3X,22HRESIDUES AT THE POLES:/,T16,9HP O L E S,T41,15HR
```

```
     1E S I D U E S,/,T9,7HREAL(A),T26,7HIMAG(B))
180   FORMAT (/,4X,1H(,F13.6,4H)+J(,F13.6,1H),4X,1H(,F13.6,1H),3A8,A1)
190   FORMAT (/,4X,1H(,F13.6,4H)+J(,F13.6,1H),4X,1H(,F13.6,1H),A4,I2,2X,
     12A8,A1)
      END
C=====================================================================
      SUBROUTINE BALANC (NM,N,A,LOW,IGH,SCALE)
      INTEGER I,J,K,L,M,N,JJ,NM,IGH,LOW,IEXC
      REAL*8 A(NM,N),SCALE(N)
      REAL*8 C,F,G,R,S,B2,RADIX
      REAL*8 DABS
      LOGICAL NOCONV
      DATA RADIX/Z4210000000000000/
C---------------------------------------------------------------------
      B2=RADIX*RADIX
      K=1
      L=N
      GO TO 60
C------------IN-LINE PROCEDURE FOR ROW AND COLUMN EXCHANGE------------
10    SCALE(M)=J
      IF (J .EQ. M) GO TO 40
      DO 20 I=1,L
      F=A(I,J)
      A(I,J)=A(I,M)
      A(I,M)=F
20    CONTINUE
      DO 30 I=K,N
      F=A(J,I)
      A(J,I)=A(M,I)
      A(M,I)=F
30    CONTINUE
40    GO TO (50,90), IEXC
C-----SEARCH FOR ROWS ISOLATING AN EIGENVALUE AND PUSH THEM DOWN--------
50    IF (L .EQ. 1) GO TO 230
      L=L-1
60    DO 80 JJ=1,L
      J=L+1-JJ
      DO 70 I=1,L
      IF (I .EQ. J) GO TO 70
      IF (A(J,I) .NE. 0.0D0) GO TO 80
70    CONTINUE
      M=L
      IEXC=1
      GO TO 10
80    CONTINUE
      GO TO 100
C-------SEARCH FOR COLUMNS ISOLATING AN EIGENVALUE AND PUSH THEM LEFT---
90    K=K+1
```

```
100   DO 120 J=K,L
      DO 110 I=K,L
      IF (I .EQ. J) GO TO 110
      IF (A(I,J) .NE. 0.0D0) GO TO 120
110   CONTINUE
      M=K
      IEXC=2
      GO TO 10
120   CONTINUE
C---------------NOW BALANCE THE SUBMATRIX IN ROWS K TO L---------------
      DO 130 I=K,L
130   SCALE(I)=1.0D0
C---------------ITERATIVE LOOP FOR NORM REDUCTION--------------------
140   NOCONV=.FALSE.
      DO 220 I=K,L
      C=0.0D0
      R=0.0D0
      DO 150 J=K,L
      IF (J .EQ. I) GO TO 150
      C=C+DABS(A(J,I))
      R=R+DABS(A(I,J))
150   CONTINUE
C---------------GUARD AGAINST ZERO C OR R DUE TO UNDERFLOW-------------
      IF (C .EQ. 0.0D0 .OR. R .EQ. 0.0D0) GO TO 220
      G=R/RADIX
      F=1.0D0
      S=C+R
160   IF (C .GE. G) GO TO 170
      F=F*RADIX
      C=C*B2
      GO TO 160
170   G=R*RADIX
180   IF (C .LT. G) GO TO 190
      F=F/RADIX
      C=C/B2
      GO TO 180
C----------------------------NOW BALANCE-----------------------------
190   IF ((C + R) / F .GE. 0.95D0 * S) GO TO 220
      G=1.0D0/F
      SCALE(I)=SCALE(I)*F
      NOCONV=.TRUE.
      DO 200 J=K,N
200   A(I,J)=A(I,J)*G
      DO 210 J=1,L
210   A(J,I)=A(J,I)*F
220   CONTINUE
      IF (NOCONV) GO TO 140
230   LOW=K
```

```
      IGH=L
      RETURN
      END
C=====================================================================
      SUBROUTINE ORTHES (NM,N,LOW,IGH,A,ORT)
      INTEGER I,J,M,N,II,JJ,LA,MP,NM,IGH,KP1,LOW
      REAL*8 A(NM,N),ORT(IGH)
      REAL*8 F,G,H,SCALE
      REAL*8 DSQRT,DABS,DSIGN
      LA=IGH-1
      KP1=LOW+1
      IF (LA .LT. KP1) GO TC 100
      DO 90 M=KP1,LA
      H=0.0D0
      ORT(M)=0.0D0
      SCALE=0.0D0
C--------------SCALE COLUMN (ALGOL TOL THEN NOT NEEDED)--------------
      DO 10 I=M,IGH
10    SCALE=SCALE+DABS(A(I,M-1))
      IF (SCALE .EQ. 0.0D0) GO TO 90
      MP=M+IGH
      DO 20 II=M,IGH
      I=MP-II
      ORT(I)=A(I,M-1)/SCALE
      H=H+ORT(I)*ORT(I)
20    CONTINUE
      G=-DSIGN(DSCRT(H),ORT(M))
      H=H-ORT(M)*G
      ORT(M)=ORT(M)-G
C----------------------FCRM (I-(U*UT)/H) * A------------------------
      DO 50 J=M,N
      F=0.0D0
      DO 30 II=M,IGH
      I=MP-II
      F=F+ORT(I)*A(I,J)
30    CONTINUE
      F=F/H
      DO 40 I=M,IGH
40    A(I,J)=A(I,J)-F*ORT(I)
50    CONTINUE
C----------------FORM (I-(U*UT)/H)*A*(I-(U*UT)/H)------------------
      DO 80 I=1,IGH
      F=0.0D0
      DO 60 JJ=M,IGH
      J=MP-JJ
      F=F+ORT(J)*A(I,J)
60    CONTINUE
      F=F/H
```

```
      DO 70 J=M,IGH
70    A(I,J)=A(I,J)-F*ORT(J)
80    CONTINUE
      ORT(M)=SCALE*ORT(M)
      A(M,M-1)=SCALE*G
90    CONTINUE
100   RETURN
      END
C=====================================================================
      SUBROUTINE ORTRAN (NM,N,LOW,IGH,A,ORT,Z)
      INTEGER I,J,N,KL,MM,MP,NM,IGH,LOW,MP1
      REAL*8 A(NM,IGH),ORT(IGH),Z(NM,N)
      REAL*8 G
C---------------INITIALIZE Z TO IDENTITY MATRIX----------------------
      DO 20 I=1,N
      DO 10 J=1,N
10    Z(I,J)=0.0D0
      Z(I,I)=1.0D0
20    CONTINUE
      KL=IGH-LOW-1
      IF (KL .LT. 1) GO TO 80
      DO 70 MM=1,KL
      MP=IGH-MM
      IF (A(MP,MP-1) .EQ. 0.0D0) GO TO 70
      MP1=MP+1
      DO 30 I=MP1,IGH
30    ORT(I)=A(I,MP-1)
      DO 60 J=MP,IGH
      G=0.0D0
      DO 40 I=MP,IGH
40    G=G+ORT(I)*Z(I,J)
C---------------DIVISOR BELOW IS NEGATIVE OF H FORMED IN ORTHES.-------
C---------------DOUBLE DIVISION AVOIDS POSSIBLE UNDERFLOW-------------
      G=(G / ORT(MP))/A(MP,MP-1)
      DO 50 I=MP,IGH
50    Z(I,J)=Z(I,J)+G*ORT(I)
60    CONTINUE
70    CONTINUE
80    RETURN
      END
C=====================================================================
      SUBROUTINE HQR2 (NM,N,LOW,IGH,H,WR,WI,Z,IERR)
      INTEGER I,J,K,L,M,N,EN,II,JJ,LL,MM,NA,NM,NN,IGH,ITS,LOW,MP2,ENM2,I
     1ERR
      REAL*8 H(NM,N),WR(N),WI(N),Z(NM,N)
      REAL*8 P,Q,R,S,T,W,X,Y,RA,SA,VI,VR,ZZ,NORM,MACHEP
      REAL*8 DSQRT,DABS,DSIGN
      INTEGER MINO
```

```
      LOGICAL NOTLAS
      COMPLEX *16Z3
      COMPLEX *16DCMPLX
      REAL*8 DREAL,DIMAG
C----------STATEMENT FUNCTICNS ENABLE EXTRACTION OF REAL AND IMAGINARY--
C----------PARTS OF DOUBLE PRECISION COMPLEX NUMBERS-------------------
      DREAL(Z3)=Z3
      DIMAG(Z3)=(0.0D0,-1.0D0)*Z3
      DATA MACHEP/Z3410000000000000/
      IERR=0
      NORM=0.0D0
      K=1
C-------STORE ROOTS ISOLATEC BY BALANC AND COMPUTE MATRIX NORM----------
      DO 20 I=1,N
      DO 10 J=K,N
 10   NORM=NORM+DABS(H(I,J))
      K=I
      IF (I .GE. LOW .AND. I .LE. IGH) GO TO 20
      WR(I)=H(I,I)
      WI(I)=0.0D0
20    CONTINUE
      EN=IGH
      T=0.0D0
C----------------SEARCH FOR NEXT EIGENVALUES----------------------------
30    IF (EN .LT. LOW) GO TC 290
      ITS=0
      NA=EN-1
      ENM2=NA-1
C----------------LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT-------------
40    DO 50 LL=LOW,EN
      L=EN+LOW-LL
      IF (L .EQ. LOW) GO TO 60
      S=DABS(H(L-1,L-1))+DAES(H(L,L))
      IF (S .EQ. 0.0D0) S=NCRM
      IF (DABS(H(L,L-1)) .LE. MACHEP * S) GO TO 60
50    CONTINUE
C-----------------------------FORM SHIFT--------------------------------
60    X=H(EN,EN)
      IF (L .EQ. EN) GO TO 220
      Y=H(NA,NA)
      W=H(EN,NA)*H(NA,EN)
      IF (L .EQ. NA) GO TO 230
      IF (ITS .EQ. 30) GO TC 500
      IF (ITS .NE. 10 .AND. ITS .NE. 20) GO TO 80
C--------------------FORM EXCEPTIONAL SHIFT-----------------------------
      T=T+X
      DO 70 I=LOW,EN
70    H(I,I)=H(I,I)-X
```

```
      S=DABS(H(EN,NA))+DABS(H(NA,ENM2))
      X=0.75D0*S
      Y=X
      W=-0.4375D0*S*S
80    ITS=ITS+1
C---------LOOK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL ELEMENTS.---------
      DO 90 MM=L,ENM2
      M=ENM2+L-MM
      ZZ=H(M,M)
      R=X-ZZ
      S=Y-ZZ
      P=(R * S - W)/H(M+1,M)+H(M,M+1)
      Q=H(M+1,M+1)-ZZ-R-S
      R=H(M+2,M+1)
      S=DABS(P)+DABS(Q)+DABS(R)
      P=P/S
      Q=Q/S
      R=R/S
      IF (M .EQ. L) GO TO 100
      IF (DABS(H(M,M-1)) * (DABS(Q) + DABS(R)) .LE. MACHEP * DABS(P)
     1 * (DABS(H(M-1,M-1)) + DABS(ZZ) + DABS(H(M+1,M+1)))) GO TO 100
90    CONTINUE
100   MP2=M+2
      DO 110 I=MP2,EN
      H(I,I-2)=0.0D0
      IF (I .EQ. MP2) GO TO 110
      H(I,I-3)=0.0D0
110   CONTINUE
C---------DOUBLE QR STEP INVOLVING ROWS L TO EN AND COLUMNS M TO EN------
      DO 210 K=M,NA
      NOTLAS=K.NE.NA
      IF (K .EQ. M) GO TO 120
      P=H(K,K-1)
      Q=H(K+1,K-1)
      R=0.0D0
      IF (NOTLAS) R=H(K+2,K-1)
      X=DABS(P)+DABS(Q)+DABS(R)
      IF (X .EQ. 0.0D0) GO TO 210
      P=P/X
      Q=Q/X
      R=R/X
120   S=DSIGN(DSQRT(P*P+Q*Q+R*R),P)
      IF (K .EQ. M) GO TO 130
      H(K,K-1)=-S*X
      GO TO 140
130   IF (L .NE. M) H(K,K-1)=-H(K,K-1)
140   P=P+S
      X=P/S
```

```
      Y=Q/S
      ZZ=R/S
      Q=Q/P
      R=R/P
C-------------------------ROW MODIFICATION-------------------------
      DO 160 J=K,N
      P=H(K,J)+Q*H(K+1,J)
      IF (.NOT. NOTLAS) GO TO 150
      P=P+R*H(K+2,J)
      H(K+2,J)=H(K+2,J)-P*ZZ
  150 H(K+1,J)=H(K+1,J)-P*Y
      H(K,J)=H(K,J)-P*X
  160 CONTINUE
      J=MINO(EN,K+3)
C-----------------------COLUMN MODIFICATION-------------------------
      DO 180 I=1,J
      P=X*H(I,K)+Y*H(I,K+1)
      IF (.NOT. NCTLAS) GO TO 170
      P=P+ZZ*H(I,K+2)
      H(I,K+2)=H(I,K+2)-P*R
  170 H(I,K+1)=H(I,K+1)-P*Q
      H(I,K)=H(I,K)-P
  180 CONTINUE
C---------------------ACCUMULATE TRANSFORMATIONS-------------------
      DO 200 I=LOW,IGH
      P=X*Z(I,K)+Y*Z(I,K+1)
      IF (.NOT. NOTLAS) GO TO 190
      P=P+ZZ*Z(I,K+2)
      Z(I,K+2)=Z(I,K+2)-P*R
  190 Z(I,K+1)=Z(I,K+1)-P*Q
      Z(I,K)=Z(I,K)-P
  200 CONTINUE
  210 CONTINUE
      GO TO 40
C--------------------------ONE ROOT FOUND----------------------------
  220 H(EN,EN)=X+T
      WR(EN)=H(EN,EN)
      WI(EN)=0.0D0
      EN=NA
      GO TO 30
C-------------------------TWO ROOTS FOUND----------------------------
  230 P=(Y - X)/2.0D0
      Q=P*P+W
      ZZ=DSQRT(DABS(Q))
      H(EN,EN)=X+T
      X=H(EN,EN)
      H(NA,NA)=Y+T
      IF (Q .LT. 0.0D0) GO TO 270
```

```
C---------------------------------REAL PAIR----------------------------------
      ZZ=P+DSIGN(ZZ,P)
      WR(NA)=X+ZZ
      WR(EN)=WR(NA)
      IF (ZZ .NE. 0.0D0) WR(EN)=X-W/ZZ
      WI(NA)=0.0D0
      WI(EN)=0.0D0
      X=H(EN,NA)
      S=DABS(X)+DABS(ZZ)
      P=X/S
      Q=ZZ/S
      R=DSQRT(P*P+Q*Q)
      P=P/R
      Q=Q/R
C------------------------------ROW MODIFICATION------------------------------
      DO 240 J=NA,N
      ZZ=H(NA,J)
      H(NA,J)=Q*ZZ+P*H(EN,J)
      H(EN,J)=Q*H(EN,J)-P*ZZ
240   CONTINUE
C------------------------------COLUMN MODIFICATION---------------------------
      DO 250 I=1,EN
      ZZ=H(I,NA)
      H(I,NA)=Q*ZZ+P*H(I,EN)
      H(I,EN)=Q*H(I,EN)-P*ZZ
250   CONTINUE
C-----------------------------ACCUMULATE TRANSFORMATIONS---------------------
      DO 260 I=LOW,IGH
      ZZ=Z(I,NA)
      Z(I,NA)=Q*ZZ+P*Z(I,EN)
      Z(I,EN)=Q*Z(I,EN)-P*ZZ
260   CONTINUE
      GO TO 280
C-----------------------------COMPLEX PAIR-----------------------------------
270   WR(NA)=X+P
      WR(EN)=X+P
      WI(NA)=ZZ
      WI(EN)=-ZZ
280   EN=ENM2
      GO TO 30
C----------------ALL ROOTS FOUND.  BACKSUBSTITUTE TO FIND--------------
C----------------VECTORS OF UPPER TRIANGULAR FORM-----------------------
290   IF (NORM .EQ. 0.0D0) GO TO 510
      DO 450 NN=1,N
      EN=N+1-NN
      P=WR(EN)
      Q=WI(EN)
      NA=EN-1
```

```
      IF (Q) 370,300,450
C---------------------------REAL VECTOR--------------------------------
300   M=EN
      H(EN,EN)=1.0D0
      IF (NA .EQ. 0) GO TO 450
      DO 360 II=1,NA
      I=EN-II
      W=H(I,[illegible]-P
      R=H(J [illegible]N)
      IF (I [illegible].GT. NA) GO TO 320
      DO [illegible] J=M,NA
310   R=[illegible] [illegible],J) *H(J,EN)
320   I[illegible] [illegible](I) .GE. 0.0D0) GO TO 330
      [illegible]
      [illegible]
      GO TO 360
330   M=I
      IF (WI(I) .NE. 0.0D0) GO TO 340
      T=W
      IF (W .EQ. 0.0D0) T=MACHEP*NORM
      H(I,EN)=-R/T
      GO TO 360
C-------------------------SOLVE REAL EQUATIONS-------------------------
340   X=H(I,I+1)
      Y=H(I+1,I)
      Q=(WR(I) - P)*(WR(I) - P)+WI(I)*WI(I)
      T=(X * S - ZZ * R)/Q
      H(I,EN)=T
      IF (DABS(X) .LE. DABS(ZZ)) GO TO 350
      H(I+1,EN)=(-R - W * T)/X
      GO TO 360
350   H(I+1,EN)=(-S - Y * T)/ZZ
360   CONTINUE
C--------------------------END REAL VECTOR-----------------------------
      GO TO 450
C--------------------------COMPLEX VECTOR------------------------------
370   M=NA
C----------------LAST VECTOR COMPONENT CHOSEN IMAGINARY SO THAT---------
C----------------EIGENVECTOR MATRIX IS TRIANGULAR-----------------------
      IF (DABS(H(EN,NA)) .LE. DABS(H(NA,EN))) GO TO 380
      H(NA,NA)=Q/H(EN,NA)
      H(NA,EN)=-(H(EN,EN) - P)/H(EN,NA)
      GO TO 390
380   Z3=DCMPLX(0.0D0,-H(NA,EN))/DCMPLX(H(NA,NA)-P,Q)
      H(NA,NA)=DREAL(Z3)
      H(NA,EN)=DIMAG(Z3)
390   H(EN,NA)=0.0D0
      H(EN,EN)=1.0D0
```

```
40    WRITE (5,130)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
      GO TO 60
50    WRITE (5,140)
      GO TO 40
60    CONTINUE
      IF (IANS.EQ.IZ) GO TO 90
      WRITE (5,150)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,160)
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,100) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 80 I=1,NO
      DO 70 J=1,NC
      IF ((I.EQ.K).AND.(J.EQ.L)) D(I,J)=DUM
70    CONTINUE
80    CONTINUE
      GO TO 30
90    CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C------------------------------------------------------------------------
100   FORMAT (5X,14HTHE ELEMENT D(,I2,1H,,I2,2H)=)
110   FORMAT (/,5X,54HENTER THE MEASUREMENT FEEDTHROUGH MATRIX / FEEDFOR
     1WARD,/,5X,34H DISTRIBUTION MATRIX {"D"-MATRIX}.,//,8X,49HDIMENSION
     2 = # OBSERVATIONS {NO} X # CONTROLS {NC})
120   FORMAT (//,5X,50HTHE FEEDFORWARD DISTRIBUTION MATRIX {"D"-MATRIX}.
     1,//)
130   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
140   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
150   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
160   FORMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1.)
      END
C========================================================================
      SUBROUTINE READG (NS,NC,ISAG,G)
C     INTERACTIVELY INPUTS THE "G" MATRIX {CONTROL DISTRIBUTION MATRIX}=
C========================================================================
      REAL*8  G(NS,NC),DUM,ANSR
      INTEGER IANS,I,J,K,L,ISAG
      DATA IY/'Y'/,IZ/'N'/
      IF (ISAG.EQ.1) GO TO 40
```

```
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,110) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 90 I=1,NO
      DO 80 J=1,NS
      IF ((I.EQ.K).AND.(J.EQ.L)) HO(I,J)=DUM
80    CONTINUE
90    CONTINUE
      GO TO 30
100   CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
110   FORMAT (5X,14HTHE ELEMENT H(,I2,1H,,I2,2H)=)
120   FORMAT (/,5X,50HENTER THE MEASUREMENT SCALING MATRIX {"H"-MATRIX}.
     1,//,10X,47HDIMENSION = # OBSERVATIONS {NO} X # STATES {NS})
130   FORMAT (//,10X,46HTHE MEASUREMENT SCALING MATRIX {"H"-MATRIX}...,/
     1/)
140   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
150   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
160   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
170   FORMAT (5X,52HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1)
      END
C=======================================================================
      SUBROUTINE READD (NO,NC,ISAD,D)
C     INPUTS THE "D" MATRIX {MEASUREMENT FEED-FORWARD DIST. MATRIX}.   =
C=======================================================================
      REAL*8  D(NC,NC),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      IF (ISAD.EQ.1) GO TO 35
      WRITE (5,110)
      DO 20 I=1,NO
      DO 10 J=1,NC
      WRITE (5,100) I,J
      CALL RDREAL (ANSR)
      D(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C-----------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
35    CONTINUE
      WRITE (5,120)
      CALL MATPRT (D,NO,NC)
```

```
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
120   FORMAT (5X,14HTHE ELEMENT F(,I2,1H,,I2,2H)=)
130   FORMAT (/,5X,36HENTER THE SYSTEM MATRIX {"F"-MATRIX},//,10X,41HDIM
     1ENSION = # STATES {NS} X # STATES {NS})
140   FORMAT (//,15X,33HTHE SYSTEM MATRIX {"F"-MATRIX}...,//)
150   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
160   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
170   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
180   FORMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1.)
      END
C=======================================================================
      SUBROUTINE READH (NO,NS,ISAH,HO)
C     INTERACTIVELY INPUTS THE "H" MATRIX {MEASUREMENT SCALING MATRIX}.=
C=======================================================================
      REAL*8  HO(NO,NS),DUM,ANSR
      INTEGER IANS,I,J,K,L,ISAH
      DATA IY/'Y'/,IZ/'N'/
C-----------------------------------------------------------------------
      IF (ISAH.EQ.1) GO TO 40
      WRITE (5,120)
      DO 20 I=1,NO
      DO 10 J=1,NS
      WRITE (5,110) I,J
      CALL RDREAL (ANSR)
      HO(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C-----------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
40    CONTINUE
      WRITE (5,130)
      CALL MATPRT (HO,NO,NS)
50    WRITE (5,140)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
      GO TO 70
60    WRITE (5,150)
      GO TO 50
70    CONTINUE
      IF (IANS.EQ.IZ) GO TO 100
      WRITE (5,160)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,170)
```

```
10    FORMAT (35H FAILURE IN HQR2 ON EIGENVALUE NO. ,I3)
      END
C=========================================================================
      SUBROUTINE READF (NS,ISAF,BA)
C     INTERACTIVELY INPUTS THE "F" MATRIX ELEMENT BY ELEMENT.            =
C=========================================================================
      REAL*8  BA(NS,NS),DUM,ANSR
      INTEGER I,J,K,L,IANS,ISAF
      DATA IY/'Y'/,IZ/'N'/
      IF (ISAF.EQ.1) GO TO 40
      WRITE (5,130)
      DO 20 I=1,NS
      DO 10 J=1,NS
      WRITE (5,120) I,J
      CALL RDREAL (ANSR)
      BA(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C-------------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
40    CONTINUE
      WRITE (5,140)
      CALL MATPRT (BA,NS,NS)
50    WRITE (5,150)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
      GO TO 70
60    WRITE (5,160)
      GO TO 50
70    CONTINUE
      IF (IANS.EQ.IZ) GO TO 110
      IF (IANS.EQ.IY) GO TO 80
80    WRITE (5,170)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,180)
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,120) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 100 I=1,NS
      DO 90 J=1,NS
      IF ((I.EQ.K).AND.(J.EQ.L)) BA(I,J)=DUM
90    CONTINUE
100   CONTINUE
      GO TO 30
110   CONTINUE
```

```
260   RE=WR(II)
      AI=WI(II)
      ZD=DCMPLX(RE**2 + AI**2 -OM**2,-2.D0*RE*OM)
      ZN=DCMPLX(RES(II+1)*AI-RES(II)*RE,RES(II)*OM)
      ZZ=ZZ+ZN/ZD
270   CONTINUE
      IF (IYU .EQ. 2 .OR. I .NE. L) GO TO 280
      PSD(K)=PSD(K)+DN1
280   PSD(K)=PSD(K)+DN1*(ZZ*DCONJG(ZZ))
290   CONTINUE
300   CONTINUE
      IF (IYU .EQ. 1) WRITE (6,390) L
      IF (IYU .EQ. 2) WRITE (6,400) L
      WRITE (6,410) (W(I),PSD(I),I=1,30)
310   CONTINUE
      RETURN
320   CONTINUE
      CALL EREXIT (N2,FA,IERR)
      RETURN
C-----------------------------------------------------------------------
330   FORMAT (/,41H SUBSEQUENT PSD IS NORMALIZED BY MEAS NO.,I3,//)
340   FORMAT (/,50H SUBSEQUENT PSD IS NORMALIZED BY PROCESS NOISE NO.,I3
     1,//)
350   FORMAT (/38H TRANSFER FUNCTION FROM PROCESS NOISE ,I2,3H TO,13H ME
     1ASUREMENT ,I2,//)
360   FORMAT (/38H TRANSFER FUNCTION FROM PROCESS NOISE ,I2,3H TO,9H CON
     1TROL ,I2,//)
370   FORMAT (/36H TRANSFER FUNCTION FROM MEASUREMENT ,I2,16H TO MEASURE
     1MENT ,I2,//)
380   FORMAT (/36H TRANSFER FUNCTION FROM MEASUREMENT ,I2,12H TO CONTROL
     1 ,I2,//)
390   FORMAT (/14H PSD OF OUTPUT,I3,32H  FORCED BY ALL NOISE-(RAD FREQ,,
     115HNORMALIZED PSD)/)
400   FORMAT (/15H PSD OF CCNTROL,I3,32H  FORCED BY ALL NOISE-(RAD FREQ,
     1,15HNORMALIZED PSD)/)
410   FORMAT (4(1X,1H(,E11.4,1H,,E11.4,1H)))
      END
C=====================================================================
      SUBROUTINE EREXIT (N,A,IERR)
C     EREXIT RETURNS THE NUMBER OF THE EIGENVALUE WHERE HQR2          =
C     FAILS, THEN STOPS THE PROGRAM.                                  =
C=====================================================================
      INTEGER IERR
      DOUBLE PRECISION A
      DIMENSION A(N,N)
      WRITE (5,10) IERR
      CALL RAPRNT (N,N,N,9,A,4,'(9(1X,1PD13.6))')
      RETURN
```

```
      DN1=DNORM*Q(I,I)
      IF (IYU .EQ. 1 .AND. IPT .EQ. 1) WRITE (6,350) I,L
      IF (IYU .EQ. 2 .AND. IPT .EQ. 1) WRITE (6,360) I,L
      IF (IYU.EQ.1) CALL RESID (I,L,N2,JCF,NG,GW,NL,HY,WR,WI,
     1RES,BB,CC,IPT)
      IF (IYU.EQ.2) CALL RESID (I,L,N2,JCF,NG,GW,NL,HU,WR,WI,
     1RES,BB,CC,IPT)
      DO 210 K=1,20
      ZZ=DCMPLX(0.D0,0.D0)
      OM=W(K)
      DO 200 II=1,N2
      IF (WI(II)) 200,180,190
180   ZD=DCMPLX(-WR(II),OM-WI(II))
      ZZ=RES(II)/ZD+ZZ
      GO TO 200
190   RE=WR(II)
      AI=WI(II)
      ZD=DCMPLX(RE**2 + AI**2 - OM**2,-2.D0*RE*OM)
      ZN=DCMPLX(RES(II+1)*AI-RES(II)*RE,RES(II)*OM)
      ZZ=ZZ+ZN/ZD
200   CONTINUE
210   PSD(K)=PSD(K)+DN1*(ZZ*DCONJG(ZZ))
220   CONTINUE
C---------------------------GSUBV-------------------------------------
      DO 240 I=1,N2
      DO 240 J=1,NO
      ST=0.D0
      DO 230 K=1,NS
230   ST=ST+X(I,K)*FBGE(K,J)+X(I,NS+K)*FBGE(K,J)
240   GV(I,J)=ST
      CALL RAPRNT (N2,N2,NO,9,GV,4,'(9(1X,1PD13.6))')
C---------------DEBUG ABOVE, LOOP THRU MEAS NOISE--------------------
      DO 300 I=1,NO
      DN1=DNORM*R(I,I)
      IF (IYU .EQ. 1 .AND. IPT .EQ. 1) WRITE (6,370) I,L
      IF (IYU .EQ. 2 .AND. IPT .EQ. 1) WRITE (6,380) I,L
      IF (IYU.EQ.1) CALL RESID (I,L,N2,JCF,NO,GV,NL,HY,WR,WI,RES,
     1      BB,CC,IPT)
      IF (IYU.EQ.2) CALL RESID (I,L,N2,JCF,NO,GV,NL,HU,WR,WI,RES,
     1      BB,CC,IPT)
      DO 290 K=1,30
      ZZ=DCMPLX(0.D0,0.D0)
      OM=W(K)
      DO 270 II=1,N2
      IF (WI(II)) 270,250,260
250   ZD=DCMPLX(-WR(II),OM-WI(II))
      ZZ=ZZ+RES(II)/ZD
      GO TO 270
```

```
      EMAX=0.D0
      DO 120 I=1,N2
      EMOD=DABS(WR(I)**2 +WI(I)**2)
      IF (EMOD .GT. EMAX) EMAX=EMOD
120   CONTINUE
      EMOD=DSQRT(EMAX)
      EMOD=2*EMOD
C--------------------ROUND UP TO NEAREST 2,4,5,8,10------------------
      ELOG=DLOG10(EMOD)
      IF (ELOG .LT. 0.D0) IPOW=-IDINT(DABS(ELOG) + 1)
      IF (ELOG .GE. 0.D0) IPOW=IDINT(ELOG)
      EMAX=EMOD*10**(-IPOW)
      IF (EMAX .GT. 2.D0) EMOD=2.D0
      IF (EMAX .GT. 4.D0) EMOD=4.D0
      IF (EMAX .GT. 5.D0) EMOD=5.D0
      IF (EMAX .GT. 8.D0) EMOD=8.D0
      IF (EMAX .GE. 10.D0) EMOD=10.D0
      EMAX=EMOD*10**IPOW
      DW=EMAX/20.D0
C---------------------ADD 10 POINTS 3 DECADES UP---------------------
      IF (EMOD .LT. 5.0) GO TO 130
      EMAX=1.0D1
      IK=3
      GO TO 140
130   EMAX=5.D0
      IK=2
140   CONTINUE
C---------------------STORE 30 FREQUENCIES---------------------------
      DO 150 I=1,20
150   W(I)=DW*(I-1)
      DO 160 I=1,3
      IP=20+3*(I-1)
      DO 160 J=1,3
      IX=MOD(IK+J-1,3)+1
      JJ=0
      IF (IK .EQ. 2 .AND. J .GE. 2) JJ=1
      W(IP+J)=DW1(IX)*10**(IPOW+I-1+JJ+IK-2)
160   CONTINUE
      IX=MOD(IK,3)+1
      W(30)=DW1(IX)*10**(IPCW+3 +IK-2)
C----------------------LARGE LOOP THRU OUTPUTS-----------------------
      IF (IYU .EQ. 1) NL=NO
      IF (IYU .EQ. 2) NL=NC
      DO 310 L=1,NL
      DO 170 I=1,30
170   PSD(I)=0.D0
C----------------------LOCP THRU PROCESS NOISE-----------------------
      DO 220 I=1,NG
```

```
      ST=0.D0
      DO 20 K=1,NO
20    ST=ST+FBGE(I,K)*H(K,J)
      FA(I,NS+J)=-ST
30    FA(NS+I,NS+J)=F(I,J)-ST
      CALL RAPRNT (N2,N2,N2,9,FA,4,'(9(1X,1PD13.6))')
C-----------------------------DEBUG ABOVE-----------------------------
      CALL BALANC (N2,N2,FA,LOW,IHIGH,D1)
      CALL ORTHES (N2,N2,LOW,IHIGH,FA,D2)
      CALL ORTRAN (N2,N2,LOW,IHIGH,FA,D2,X)
      CALL HQR2 (N2,N2,LOW,IHIGH,FA,WR,WI,X,IERR)
      IF (IERR .NE. 0) GO TC 320
      CALL BALBAK (N2,N2,LOW,IHIGH,D1,N2,X)
      CALL RAPRNT (N2,N2,N2,9,X,4,'(9(1X,1PD13.6))')
C------------DEBUG ABOVE;DETERMINE MODAL MATRICES---------------------
      IF (IYU .EQ. 1) GO TO 60
C----------------------------HSUBU------------------------------------
      DO 50 I=1,NC
      DO 50 J=1,N2
      ST=0.D0
      DO 40 K=1,NS
40    ST=ST-C(I,K)*X(K,J)
50    HU(I,J)=ST
      GO TO 90
C----------------------------HSUBY------------------------------------
60    DO 80 I=1,NO
      DO 80 J=1,N2
      ST=0.D0
      DO 70 K=1,NS
70    ST=ST+H(I,K)*X(K,J)-H(I,K)*X(NS+K,J)
80    HY(I,J)=ST
      CALL RAPRNT (NO,NO,N2,9,HY,4,'(9(1X,1PD13.6))')
C----------------------------DEBUG ABOVE------------------------------
90    CALL MINV (NSQ,X,N2,ST,D1,D2)
      CALL RAPRNT (N2,N2,N2,9,X,4,'(9(1X,1PD13.6))')
C----------------------------DEBUG ABOVE------------------------------
C----------------------------GSUBW------------------------------------
      DO 110 I=1,N2
      DO 110 J=1,NG
      ST=0.0D0
      DO 100 K=1,NS
100   ST=ST-X(I,NS+K)*GAM(K,J)
110   GW(I,J)=ST
      CALL RAPRNT (N2,N2,NG,9,GW,4,'(9(1X,1PD13.6))')
C---------DEBUG ABOVE; USE SELECTED NORMALIZATION---------------------
      IF (INORM .LE. NG) DNCRM=1.D0/Q(INORM,INORM)
      IF (INORM .GT. NG) DNORM=1.D0/R(INORM-NG,INORM-NG)
C---------------DETERMINE EANDWIDTH OF CONTROLLED SYSTEM--------------
```

```
C----------------SET ERROR -- NO CONVERGENCE TO AN---------------------
C----------------EIGENVALUE AFTER 30 ITERATIONS------------------------
240   IERR=EN
250   RETURN
      END
C======================================================================
      SUBROUTINE PSDCAL (N2,NS,FA,X,NC,GW,GV ,C,NO,HY,HU,H,
     1  FBGE,NG,GAM,ACL,F,WR,WI,D1,D2,JCF,RES,Q,R,BB,CC,IYU,
     2  IPSD,INORM)
C======================================================================
C     =  PSDCAL COMPUTES THE PSD OF OUTPUTS OR CONTROLS OF            =
C     =   A CONTROLLED SYSTEM                                         =
C     =                                                               =
C     =     IYU= 1       OUTPUT PSD                                   =
C     =        = 2       CONTROL PSD                                  =
C     =        = 3       BOTH OUTPUT AND CONTROL PSD                  =
C     =                                                               =
C     =     IPSD=1       PSD                                          =
C     =         =2       PSD AND TF RESIDUES                          =
C     =                                                               =
C     =                                                               =
C     =     INORM=       1,2,... NG  NORMALIZED BY ITH PROCESS NOISE  =
C     =                  NG+1,... NG+NO NORMALIZED BY ITH MEAS NOISE  =
C======================================================================
      DOUBLE PRECISION FA,X,GW,GV,C,HY,H,FBGE,GAM,ACL,F,WR,WI,D1,D2,RES,
     1BB,CC,Q,R,PSD,W,DNORM,DN1,EMAX,ELOG,EMOD,DW,ST,OM,RE,AI,HU,DW1
      COMPLEX *16ZD,ZN,ZZ
      DIMENSION FA(N2,N2),X(N2,N2),GW(N2,NG),C(NC,NS),HY(NO,N2),H(NO,NS)
     1,FBGE(NS,NO),GAM(NS,NC),ACL(NS,NS),F(NS,NS),WR(N2),WI(N2),D1(N2),D
     22(N2),RES(N2),Q(NG,NG),R(NO,NO),PSD(30),W(30),BB(N2),CC(N2),GV(N2,
     3NO),HU(NC,N2),DW1(4)
      INTEGER JCF(N2)
      DATA DW1/1.D0,2.D0,5.D0,10.D0/
      IF (IYU .EQ. 0) IYU=1
      IF (INORM .EQ. 0) INORM=1
      IPT=0
      IF (IPSD .GT. 1) IPT=1
      IX=INORM-NG
      IF (IX .GT. 0) WRITE (6,330) IX
      IF (IX .LE. 0) WRITE (6,340) INORM
      NSQ=N2*N2
C--------COMPUTE EIGENSYSTEM OF CONTROLLED SYSTEM; FORM FA------------
      DO 10 I=1,NS
      DO 10 J=1,NS
      FA(I,J)=ACL(I,J)
10    FA(NS+I,J)=0.D0
      DO 30 I=1,NS
      DO 30 J=1,NS
```

```
      R=R/P
C-----------------------------ROW MODIFICATION--------------------------
      DO 160 J=K,EN
      P=H(K,J)+Q*H(K+1,J)
      IF (.NOT. NOTLAS) GO TO 150
      P=P+R*H(K+2,J)
      H(K+2,J)=H(K+2,J)-P*ZZ
150   H(K+1,J)=H(K+1,J)-P*Y
      H(K,J)=H(K,J)-P*X
160   CONTINUE
      J=MIN0(EN,K+3)
C-----------------------------COLUMN MODIFICATION-----------------------
      DO 180 I=L,J
      P=X*H(I,K)+Y*H(I,K+1)
      IF (.NOT. NOTLAS) GO TO 170
      P=P+ZZ*H(I,K+2)
      H(I,K+2)=H(I,K+2)-P*R
170   H(I,K+1)=H(I,K+1)-P*Q
      H(I,K)=H(I,K)-P
180   CONTINUE
190   CONTINUE
      GO TO 40
C-----------------------------ONE ROOT FOUND----------------------------
200   WR(EN)=X+T
      WI(EN)=0.0D0
      EN=NA
      GO TO 30
C-----------------------------TWO ROOTS FOUND---------------------------
210   P=(Y - X)/2.0D0
      Q=P*P+W
      ZZ=DSQRT(DABS(Q))
      X=X+T
      IF (Q .LT. 0.0D0) GO TO 220
C-----------------------------REAL PAIR---------------------------------
      ZZ=P+DSIGN(ZZ,P)
      WR(NA)=X+ZZ
      WR(EN)=WR(NA)
      IF (ZZ .NE. 0.0D0) WR(EN)=X-W/ZZ
      WI(NA)=0.0D0
      WI(EN)=0.0D0
      GO TO 230
C-----------------------------COMPLEX PAIR------------------------------
220   WR(NA)=X+P
      WR(EN)=X+P
      WI(NA)=ZZ
      WI(EN)=-ZZ
230   EN=ENM2
      GO TO 30
```

```
      W=-0.4375D0*S*S
80    ITS=ITS+1
C----------LOOK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL ELEMENTS.--------
      DO 90 MM=L,ENM2
      M=ENM2+L-MM
      ZZ=H(M,M)
      R=X-ZZ
      S=Y-ZZ
      P=(R * S - W)/H(M+1,M)+H(M,M+1)
      Q=H(M+1,M+1)-ZZ-R-S
      R=H(M+2,M+1)
      S=DABS(P)+DABS(Q)+DABS(R)
      P=P/S
      Q=Q/S
      R=R/S
      IF (M .EQ. L) GO TO 100
      IF (DABS(H(M,M-1)) * (DABS(Q) + DABS(R)) .LE. MACHEP * DABS(P)
     1 * (DABS(H(M-1,M-1)) + DABS(ZZ) + DABS(H(M+1,M+1)))) GO TO 100
90    CONTINUE
100   MP2=M+2
      DO 110 I=MP2,EN
      H(I,I-2)=0.0D0
      IF (I .EQ. MP2) GO TO 110
      H(I,I-3)=0.0D0
110   CONTINUE
C-------DOUBLE QR STEP INVOLVING ROWS L TO EN AND COLUMNS M TO EN--------
      DO 190 K=M,NA
      NOTLAS=K.NE.NA
      IF (K .EQ. M) GO TO 120
      P=H(K,K-1)
      Q=H(K+1,K-1)
      R=0.0D0
      IF (NOTLAS) R=H(K+2,K-1)
      X=DABS(P)+DABS(Q)+DABS(R)
      IF (X .EQ. 0.0D0) GO TO 190
      P=P/X
      Q=Q/X
      R=R/X
120   S=DSIGN(DSQRT(P*P+Q*Q+R*R),P)
      IF (K .EQ. M) GO TO 130
      H(K,K-1)=-S*X
      GO TO 140
130   IF (L .NE. M) H(K,K-1)=-H(K,K-1)
140   P=P+S
      X=P/S
      Y=Q/S
      ZZ=R/S
      Q=Q/P
```

```
      REAL*8 H(NM,N),WR(N),WI(N)
      REAL*8 P,Q,R,S,T,W,X,Y,ZZ,NORM,MACHEP
      REAL*8 DSQRT,DABS,DSIGN
      INTEGER MINO
      LOGICAL NOTIAS
      DATA MACHEP/Z341000000000000/
      IERR=0
      NORM=0.0D0
      K=1
C--------STORE ROOTS ISOLATED BY BALANC AND COMPUTE MATRIX NORM---------
      DO 20 I=1,N
      DO 10 J=K,N
10    NORM=NORM+DABS(H(I,J))
      K=I
      IF (I .GE. LOW .AND. I .LE. IGH) GO TO 20
      WR(I)=H(I,I)
      WI(I)=0.0D0
20    CONTINUE
      EN=IGH
      T=0.0D0
C-----------------------SEARCH FOR NEXT EIGENVALUES--------------------
30    IF (EN .LT. LOW) GO TC 250
      ITS=0
      NA=EN-1
      ENM2=NA-1
C---------------LOOK FOR SINGLE SMALL SUB-DIAGONAL ELEMENT-------------
40    DO 50 LL=LOW,EN
      L=EN+LOW-LL
      IF (L .EQ. LOW) GO TO 60
      S=DABS(H(L-1,L-1))+DABS(H(L,L))
      IF (S .EQ. 0.0D0) S=NCRM
      IF (DABS(H(L,L-1)) .LE. MACHEP * S) GO TO 60
50    CONTINUE
C----------------------------FORM SHIFT--------------------------------
60    X=H(EN,EN)
      IF (L .EQ. EN) GO TO 200
      Y=H(NA,NA)
      W=H(EN,NA)*H(NA,EN)
      IF (L .EQ. NA) GO TO 210
      IF (ITS .EQ. 30) GO TC 240
      IF (ITS .NE. 10 .AND. ITS .NE. 20) GO TO 80
C------------------------FORM EXCEPTIONAL SHIFT------------------------
      T=T+X
      DO 70 I=LOW,EN
70    H(I,I)=H(I,I)-X
      S=DABS(H(EN,NA))+DABS(H(NA,ENM2))
      X=0.75D0*S
      Y=X
```

```
C---------------MULTIPLY BY TRANSFORMATION MATRIX TO GIVE--------------
C---------------VECTORS OF ORIGINAL FULL MATRIX.-----------------------
      DO 490 JJ=LCW,N
      J=N+LOW-JJ
      M=MINO(J,IGH)
      DO 490 I=LOW,IGH
      ZZ=0.0D0
      DO 480 K=LOW,M
480   ZZ=ZZ+Z(I,K)*H(K,J)
      Z(I,J)=ZZ
490   CONTINUE
      GO TO 510
C---------------SET ERROR -->NO CONVERGENCE TO AN----------------------
C---------------EIGENVALUE AFTER 30 ITERATIONS-------------------------
500   IERR=EN
510   RETURN
      END
C======================================================================
      SUBROUTINE BALBAK (NM,N,LOW,IGH,SCALE,M,Z)
      INTEGER I,J,K,M,N,II,NM,IGH,LOW
      REAL*8 SCALE(N),Z(NM,M),S
      IF (M .EQ. 0) GO TO 60
      IF (IGH .EQ. LOW) GO TO 30
      DO 20 I=LOW,IGH
      S=SCALE(I)
C---------------LEFT HAND EIGENVECTORS ARE BACK TRANSFORMED------------
C---------------IF THE FOREGOING STATEMENT IS REPLACED BY--------------
C---------------S=1.0D0/SCALE(I).--------------------------------------
      DO 10 J=1,M
10    Z(I,J)=Z(I,J)*S
20    CONTINUE
30    DO 50 II=1,N
      I=II
      IF (I .GE. IOW .AND. I .LE. IGH) GO TO 50
      IF (I .LT. LOW) I=LOW-II
      K=SCALE(I)
      IF (K .EQ. I) GO TO 50
      DO 40 J=1,M
      S=Z(I,J)
      Z(I,J)=Z(K,J)
      Z(K,J)=S
40    CONTINUE
50    CONTINUE
60    RETURN
      END
C======================================================================
      SUBROUTINE HOR (NM,N,IOW,IGH,H,WR,WI,IERR)
      INTEGER I,J,K,L,M,N,EN,LL,MM,NA,NM,IGH,ITS,LOW,MP2,ENM2,IERR
```

```
      ENM2=NA-1
      IF (ENM2 .EQ. 0) GO TO 450
      DO 440 II=1,ENM2
      I=NA-II
      W=H(I,I)-P
      RA=0.0D0
      SA=H(I,EN)
      DO 400 J=M,NA
      RA=RA+H(I,J)*H(J,NA)
      SA=SA+H(I,J)*H(J,EN)
400   CONTINUE
      IF (WI(I) .GE. 0.0D0) GO TO 410
      ZZ=W
      R=RA
      S=SA
      GO TO 440
410   M=I
      IF (WI(I) .NE. 0.0D0) GO TO 420
      Z3=DCMPLX(-RA,-SA)/DCMPLX(W,Q)
      H(I,NA)=DREAL(Z3)
      H(I,EN)=DIMAG(Z3)
      GO TO 440
C-----------------------SCLVE COMPLEX EQUATIONS---------------------
420   X=H(I,I+1)
      Y=H(I+1,I)
      VR=(WR(I) - P)*(WR(I) - P)+WI(I)*WI(I)-Q*Q
      VI=(WR(I) - P)*2.0D0*Q
      IF (VR .EQ. 0.0D0 .AND. VI .EQ. 0.0D0) VR=MACHEP*NORM*(DABS(W) + D
     1ABS(Q) + DABS(X) + DABS(Y) + DABS(ZZ))
      Z3=DCMPLX(X*R-ZZ*RA+Q*SA,X*S-ZZ*SA-Q*RA)/DCMPLX(VR,VI)
      H(I,NA)=DREAL(Z3)
      H(I,EN)=DIMAG(Z3)
      IF (DABS(X) .LE. DABS(ZZ) + DABS(Q)) GO TO 430
      H(I+1,NA)=(-RA - W * H(I,NA) + Q * H(I,EN))/X
      H(I+1,EN)=(-SA - W * H(I,EN) - Q * H(I,NA))/X
      GO TO 440
430   Z3=DCMPLX(-R-Y*H(I,NA),-S-Y*H(I,EN))/DCMPLX(ZZ,Q)
      H(I+1,NA)=DREAL(Z3)
      H(I+1,EN)=DIMAG(Z3)
440   CONTINUE
C---------------------------END COMPLEX VECTOR----------------------
450   CONTINUE
C-------------END BACK SUBSTITUTION. VECTORS OF ISOLATED ROOTS---------
      DO 470 I=1,N
      IF (I .GE. LOW .AND. I .LE. IGH) GO TO 470
      DO 460 J=I,N
460   Z(I,J)=H(I,J)
470   CONTINUE
```

```
      WRITE (5,120)
      DO 20 I=1,NS
      DO 10 J=1,NC
      WRITE (5,110) I,J
      CALL RDREAL (ANSR)
      G(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C----------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
40    CONTINUE
      WRITE (5,130)
      CALL MATPRT (G,NS,NC)
50    WRITE (5,140)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
      GO TO 70
60    WRITE (5,150)
      GO TO 50
70    CONTINUE
      IF (IANS.EQ.IZ) GO TO 100
      WRITE (5,160)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,170)
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,110) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 90 I=1,NS
      DO 80 J=1,NC
      IF ((I.EQ.K).AND.(J.EQ.L)) G(I,J)=DUM
80    CONTINUE
90    CONTINUE
      GO TO 30
100   CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C----------------------------------------------------------------------
110   FORMAT (5X,14HTHE ELEMENT G(,I2,1H,,I2,2H)=)
120   FORMAT (/,5X,51HENTER THE CONTROL DISTRIBUTION MATRIX {"G"-MATRIX}
     1.,//,10X,43HDIMENSION = # STATES {NS} X # CONTROLS {NC})
130   FORMAT (//,10X,47HTHE CONTROL DISTRIBUTION MATRIX {"G"-MATRIX}...,
     1//)
140   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
150   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
```

```
160   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
170   FORMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1.)
      END
C=======================================================================
      SUBROUTINE READFB (NC,NS,FBGC)
C     INPUTS THE "C"    {FEEDBACK GAIN CONTROL MATRIX}.                =
C=======================================================================
      REAL*8  FBGC(NC,NS),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      WRITE (5,110)
      DO 20 I=1,NC
      DO 10 J=1,NS
      WRITE (5,100) I,J
      CALL RDREAL (ANSR)
      FBGC(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C-----------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,120)
      CALL MATPRT (FBGC,NC,NS)
40    WRITE (5,130)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
      GO TO 60
50    WRITE (5,140)
      GO TO 40
60    CONTINUE
      IF (IANS.EQ.IZ) GO TO 90
      WRITE (5,150)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,160)
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,100) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 80 I=1,NC
      DO 70 J=1,NS
      IF ((I.EQ.K).AND.(J.EQ.L)) FBGC(I,J)=DUM
70    CONTINUE
80    CONTINUE
      GO TO 30
90    CONTINUE
      CALL FRTCMS ('CLRSCRN ')
```

```
      RETURN
C-----------------------------------------------------------------------
100   FORMAT (5X,14HTHE ELEMENT C(,I2,1H,,I2,2H)=)
110   FORMAT (/,5X,52HENTER THE FEEDBACK GAIN CONTROL MATRIX {"C"-MATRIX
     1}.,//,10X,44HDIMENSION = # CONTROLS {NC} X # STATES {NS}.)
120   FORMAT (//,10X,45HTHE FEEDBACK GAIN CONTROL MATRIX {"C"-MATRIX},//
     1)
130   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
140   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
150   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
160   FORMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1.)
      END
C=======================================================================
      SUBROUTINE READAY (NO,ISAA,AY)
C     INPUTS THE "A" MATRIX  {DIAGONAL OUTPUT COST MATRIX}.           =
C=======================================================================
      REAL*8  AY(NO,NO),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      IF (ISAA.EQ.1) GO TO 30
      WRITE (5,110)
      DO 20 I=1,NO
      DO 10 J=1,NC
      WRITE (5,100) I,J
      CALL RDREAL (ANSR)
      AY(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C-----------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,120)
      CALL MATPRT (AY,NO,NO)
40    WRITE (5,130)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
      GO TO 60
50    WRITE (5,140)
      GO TO 40
60    CONTINUE
      IF (IANS.EQ.IZ) GO TO 90
      WRITE (5,150)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,160)
      CALL RDINT (IANS)
      L=IANS
```

```
      WRITE (5,100) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 80 I=1,NO
      DO 70 J=1,NC
      IF ((I.EQ.K).AND.(J.EQ.L)) AY(I,J)=DUM
70    CONTINUE
80    CONTINUE
      GO TO 30
90    CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
100   FORMAT (5X,14HTHE ELEMENT A(,I2,1H,,I2,2H)=)
110   FORMAT (//,5X,54HENTER THE OUTPUT MEASUREMENT COST MATRIX {"A"-MAT
     1RIX}.,/,5X,53HDIMENSICN = # OBSERVATIONS {NO} X # OBSERVATIONS {NO
     2})
120   FORMAT (//,5X,50HTHE CUTPUT MEASUREMENT COST MATRIX {"A"-MATRIX}..
     1.,//)
130   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
140   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
150   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
160   FORMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1.)
      END
C=======================================================================
      SUBROUTINE READB (NC,ISAB,B)
C     INPUTS THE "B" MATRIX {CONTROL COST WEIGHTING MATRIX}.           =
C=======================================================================
      REAL*8  B(NC,NC),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      IF (ISAB.EQ.1) GO TO 20
      WRITE (5,90)
      DO 10 I=1,NC
      DO 10 J=1,NC
      WRITE (5,80) I,J
      CALL RDREAL (ANSR)
10    B(I,J)=ANSR
C-----------------------------------------------------------------------
20    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,100)
      CALL MATPRT (B,NC,NC)
30    WRITE (5,110)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 40
      GO TO 50
```

```
40    WRITE (5,120)
      GO TO 30
50    CONTINUE
      IF (IANS.EQ.IZ) GO TO 70
      WRITE (5,130)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,140)
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,80) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 60 I=1,NC
      DO 60 J=1,NC
      IF ((I.EQ.K).AND.(J.EQ.L)) B(I,J)=DUM
60    CONTINUE
      GO TO 20
70    CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------
80    FORMAT (5X,14HTHE ELEMENT B(,I2,1H,,I2,2H)=)
90    FORMAT (/,5X,52HENTER THE CONTROL COST WEIGHTING MATRIX {"B"-MATRI
     1X},/,10X,45HDIMENSION = # CONTROLS {NC} X # CONTROLS {NC})
100   FORMAT (//,10X,37HTHE CONTROL COST MATRIX..........B...,//)
110   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
120   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
130   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
140   FORMAT (5X,52HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1)
      END
C=================================================================
      SUBROUTINE READG2 (NS,NG,IGAM,GAM)
C     INPUTS THE "GAM" MATRIX {PROCESS NOISE DISTRIBUTION MATRIX}.     =
C=================================================================
      REAL*8  GAM(NS,NG),DUM,ANSR
      INTEGER IANS,I,J,K,L,IGAM
      DATA IY/'Y'/,IZ/'N'/
      IF (IGAM.EQ.1) GO TO 40
      WRITE (5,120)
      DO 20 I=1,NS
      DO 10 J=1,NG
      WRITE (5,110) I,J
      CALL RDREAL (ANSR)
      GAM(I,J)=ANSR
10    CONTINUE
```

```
20    CONTINUE
C-----------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
40    CONTINUE
      WRITE (5,130)
      CALL MATPRT (GAM,NS,NG)
50    WRITE (5,140)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
      GO TO 70
60    WRITE (5,150)
      GO TO 50
70    CONTINUE
      IF (IANS.EQ.IZ) GO TO 100
      WRITE (5,160)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,170)
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,110) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 90 I=1,NS
      DO 80 J=1,NG
      IF ((I.EQ.K).AND.(J.EQ.L)) GAM(I,J)=DUM
80    CONTINUE
90    CONTINUE
      GO TO 30
100   CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
110   FORMAT (5X,16HTHE ELEMENT GAM(,I2,1H,,I2,2H)=)
120   FORMAT (/,5X,36HENTER THE PROCESS NOISE DISTRIBUTION,/,5X,24HMATRI
     1X {"GAMMA"-MATRIX}.,/,2X,56HDIMENSION = # STATES {NS} X # PROCESS
     2NOISE SOURCES {NG})
130   FORMAT (//,10X,37HTHE PROCESS NOISE DISTRIBUTION MATRIX,/,10X,19H{
     1"GAMMA"-MATRIX}...,//)
140   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
150   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
160   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
170   FORMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1.)
      END
C=======================================================================
      SUBROUTINE READQ (NG,Q)
```

```
C     INTERACTIVELY INPUTS THE "Q" MATRIX {NOISE WEIGHTING MATRIX}      =
C===========================================================================
      REAL*8  Q(NG,NG),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      WRITE (5,110)
      DO 20 I=1,NG
      DO 10 J=1,NG
      WRITE (5,100) I,J
      CALL RDREAL (ANSR)
      Q(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C---------------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,120)
      CALL MATPRT (Q,NG,NG)
40    WRITE (5,130)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
      GO TO 60
50    WRITE (5,140)
      GO TO 40
60    CONTINUE
      IF (IANS.EQ.IZ) GO TO 90
      WRITE (5,150)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,160)
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,100) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 80 I=1,NG
      DO 70 J=1,NG
      IF ((I.EQ.K).AND.(J.EQ.L)) Q(I,J)=DUM
70    CONTINUE
80    CONTINUE
      GO TO 30
90    CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C---------------------------------------------------------------------------
100   FORMAT (5X,14HTHE ELEMENT Q(,I2,1H,,I2,2H)=)
110   FORMAT (//,5X,44HENTER THE PROCESS NOISE PSD WEIGHTING MATRIX,/,5X
     1,12H{"Q"MATRIX}.,//,5X,42HDIMENSION = # PROCESS NOISE SOURCES {NG}
     2 X,/,17X,27H#PROCESS NOISE SOURCES {NG})
```

```
120   FORMAT (//,5X,42HTHE PROCESS NOISE WEIGHTING MATRIX.....Q...//)
130   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
140   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
150   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
160   FORMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1.)
      END
C======================================================================
      SUBROUTINE READR (NO,RC)
C     INTERACTIVELY INPUTS THE "R" MATRIX=
C     {MEASUREMENT NOISE DISTRIBUTION MATRIX.}                          =
C======================================================================
      REAL*8  RC(NO,NO),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      WRITE (5,90)
      DO 10 I=1,NC
      DO 10 J=1,NC
      WRITE (5,80) I,J
      CALL RDREAL (ANSR)
10    RC(I,J)=ANSR
C----------------------------------------------------------------------
20    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,100)
      CALL MATPRT (RC,NO,NO)
30    WRITE (5,110)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 40
      GO TO 50
40    WRITE (5,120)
      GO TO 30
50    CONTINUE
      IF (IANS.EQ.IZ) GO TO 70
      WRITE (5,130)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,140)
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,80) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 60 I=1,NO
      DO 60 J=1,NO
60    IF ((I.EQ.K).AND.(J.EQ.L)) RC(I,J)=DUM
      GO TO 20
70    CONTINUE
```

```
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
80    FORMAT (5X,14HTHE ELEMENT R(,I2,1H,,I2,2H)=)
90    FORMAT (/,5X,60HENTER THE MEASUREMENT NOISE DISTRIBUTION MATRIX {"
     1R"MATRIX}.,//,5X,53HDIMENSION = # OBSERVATIONS {NO} X # OBSERVATIO
     2NS {NO})
100   FORMAT (//,15X,50HTHE MEASUREMENT NOISE DISTRIBUTION MATRIX.....R.
     1..,//)
110   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
120   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
130   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
140   FORMAT (5X,52HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1)
      END
C=======================================================================
      SUBROUTINE READFE (NS,NO,FBGE)
C     INTERACTIVELY INPUTS THE "K" {FEEDBACK GAIN ESTIMATOR MATRIX}    =
C=======================================================================
      REAL*8  FBGE(NS,NO),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA IY/'Y'/,IZ/'N'/
      WRITE (5,110)
      DO 20 I=1,NS
      DO 10 J=1,NO
      WRITE (5,100) I,J
      CALL RDREAL (ANSR)
      FBGE(I,J)=ANSR
10    CONTINUE
20    CONTINUE
C-----------------------------------------------------------------------
30    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,120)
      CALL MATPRT (FBGE,NS,NO)
40    WRITE (5,130)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
      GO TO 60
50    WRITE (5,140)
      GO TO 40
60    CONTINUE
      IF (IANS.EQ.IZ) GO TO 90
      WRITE (5,150)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,160)
      CALL RDINT (IANS)
```

```
      L=IANS
      WRITE (5,100) K,L
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 80 I=1,NS
      DO 70 J=1,NO
      IF ((I.EQ.K).AND.(J.EQ.L)) FBGE(I,J)=DUM
70    CONTINUE
80    CONTINUE
      GO TO 30
90    CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
100   FORMAT (5X,14HTHE ELEMENT K(,I2,1H,,I2,2H)=)
110   FORMAT (/,5X,54HENTER THE FEEDBACK GAIN ESTIMATOR MATRIX {"K"-MATR
     1IX}.,//,10X,48HDIMENSION = # STATES {NS} X # OBSERVATIONS {NO}.)
120   FORMAT (//,15X,47HTHE FEEDBACK GAIN ESTIMATOR MATRIX {"K"-MATRIX},
     1//)
130   FORMAT (//,5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELE
     1MENT?,//,10X,19HTYPE "YES" OR "NO".)
140   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
150   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
160   FORMAT (5X,52HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
     1)
      END
C=======================================================================
      SUBROUTINE READW (NG,WR)
C     INTERACTIVELY INPUTS THE "WO" MATRIX {STEADY DISTURBANCE VECTOR =
C     MATRIX} ELEMENT BY ELEMENT.                                      =
C=======================================================================
      REAL*8  WR(NG),DUM,ANSR
      INTEGER IANS,I,K
      DATA IY/'Y'/,IZ/'N'/
      WRITE (5,100)
      DO 10 I=1,NG
      WRITE (5,80) I
      CALL RDREAL (ANSR)
      WR(I)=ANSR
10    CONTINUE
C-----------------------------------------------------------------------
20    CALL FRTCMS ('CLRSCRN ')
      WRITE (5,110)
      WRITE (5,90) (WR(I),I=1,NG)
30    WRITE (5,120)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 40
      GO TO 50
```

```
40    WRITE (5,130)
      GO TO 30
50    CONTINUE
      IF (IANS.EQ.IZ) GO TO 70
      WRITE (5,140)
      CALL RDINT (IANS)
      K=IANS
      WRITE (5,80) K
      CALL RDREAL (ANSR)
      DUM=ANSR
      DO 60 I=1,NG
      IF (I.EQ.K) WR(I)=DUM
60    CONTINUE
      GO TO 20
70    CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----------------------------------------------------------------------
80    FORMAT (5X,15HTHE ELEMENT W0(,I2,2H)=)
90    FORMAT (F12.5)
100   FORMAT (/,5X,57HENTER THE STEADY DISTURBANCE VECTOR MATRIX {"W0"-M
     1ATRIX}.,//,10X,44HDIMENSION = # PROCESS NOISE SOURCES {NG} X 1)
110   FORMAT (//,15X,53HTHE STEADY DISTURBANCE VECTOR MATRIX {"W0"-MATRI
     1X}...,//)
120   FORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEM
     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
130   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
140   FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
      END
C=======================================================================
C   SUBROUTINE RDREAL -- INTERACTIVELY READS A REAL NUMBER REPLY       =
C   INTO A FORTRAN PROGRAM.  IF THE USER INADVERTENLY ENTERS  A NULL   =
C   STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.             =
C=======================================================================
      SUBROUTINE RDREAL (ANSR)
      REAL*8 ANSR
      INTEGER COUNT
C-----------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      READ (5,*,END=30,ERR=30) ANSR
      RETURN
30    REWIND 5
```

```
      WRITE (5,50)
      GO TO 10
40    CONTINUE
      STOP
C-----------------------------------------------------------------------
50    FORMAT (1X,64HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER A NUME
     1RICAL VALUE.)
60    FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED!)
      END
C=======================================================================
C  SUBROUTINE RDINT -- INTERACTIVELY READS AN INTEGER REPLY            =
C  INTO A FORTRAN PROGRAM.  IF THE USER INADVERTENLY ENTERS AN IMPROPER=
C  DATA CHARACTER THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.      =
C=======================================================================
      SUBROUTINE RDINT (IANS)
      INTEGER COUNT,IANS
C-----------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 50
20    CONTINUE
      READ (5,*,END=40,ERR=40) IANS
      IF (IANS) 40,40,30
30    CONTINUE
      RETURN
40    REWIND 5
      WRITE (5,70)
      GO TO 10
50    CONTINUE
      STOP
C-----------------------------------------------------------------------
60    FORMAT (//,5X,49HPROGRAM TERMINATION - TWO IMPROPER DATA ENTRIES!!
     1)
70    FORMAT (1X,56HWARNING: IMPROPER DATA ENTRY!  ENTER A POSITIVE INTE
     1GER.)
      END
C=======================================================================
C  SUBROUTINE RDCHAR -- INTERACTIVELY READS A CHARACTER STRING REPLY   =
C  ('YES' OR 'NO') INTO A FORTRAN PROGRAM. IF THE USER INADVERTENLY    =
C  ENTERS  A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY=
C=======================================================================
      SUBROUTINE RDCHAR (IANS)
      INTEGER COUNT,IANS
C     DATA IY/'Y'/,IZ/'N'/
C-----------------------------------------------------------------------
```

```
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      REWIND 5
      READ (5,70,END=30,ERR=30) IANS
      RETURN
30    REWIND 5
      WRITE (5,50)
      GO TO 10
40    CONTINUE
      STOP
C-----------------------------------------------------------------------
50    FORMAT (1X,60HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER "YES"
     1OR "NO".)
60    FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED!)
70    FORMAT (A1)
      END
C=======================================================================
C  SUBROUTINE MATPRT -- DISPLAYS A TWO-DIMENSIONAL ARRAY (16 COLS. MAX)=
C  IN VARIABLE SCREEN FORMAT FOR USER EASE IN ROW IDENTIFICATION.      =
C=======================================================================
      SUBROUTINE MATPRT (PRTT,NROW,NCOL)
      IMPLICIT REAL*8 (A-H,C-Z)
      DIMENSION PRTT(NROW,NCOL)
C-----------------------------------------------------------------------
      IF (NCOL.EQ.0)  NCOL=1
      IF (NCOL.EQ.1)  WRITE (5,10)   ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.2)  WRITE (5,20)   ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.3)  WRITE (5,30)   ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.4)  WRITE (5,40)   ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.5)  WRITE (5,50)   ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.6)  WRITE (5,60)   ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.7)  WRITE (5,70)   ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.8)  WRITE (5,80)   ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.9)  WRITE (5,90)   ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.10) WRITE (5,100)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.11) WRITE (5,110)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.12) WRITE (5,120)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.13) WRITE (5,130)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.14) WRITE (5,140)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.15) WRITE (5,150)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      IF (NCOL.EQ.16) WRITE (5,160)  ((PRTT(I,J),J=1,NCOL),I=1,NROW)
      RETURN
C-----------------------------------------------------------------------
```

```
10     FORMAT (F12.5)
20     FORMAT (2F12.5)
30     FORMAT (3F12.5)
40     FORMAT (4F12.5)
50     FORMAT (5F12.5)
60     FORMAT (6F12.5)
70     FORMAT (6F12.5,/,F12.5,//)
80     FORMAT (6F12.5,/,2F12.5,//)
90     FORMAT (6F12.5,/,3F12.5,//)
100    FORMAT (6F12.5,/,4F12.5,//)
110    FORMAT (6F12.5,/,5F12.5,//)
120    FORMAT (6F12.5,/,6F12.5,//)
130    FORMAT (6F12.5,/,6F12.5,/,F12.5,//)
140    FORMAT (6F12.5,/,6F12.5,/,2F12.5,//)
150    FORMAT (6F12.5,/,6F12.5,/,3F12.5,//)
160    FORMAT (6F12.5,/,6F12.5,/,4F12.5,//)
       END
C=====================================================================
C  SUBROUTINE RDMATF -- READS THE FLAGS AND MATRIX SIZES FROM          =
C  THE DATA FILE CN FILEDEF 9.  ASKS IF YOU WANT TO USE THE MATRICES.  =
C=====================================================================
       SUBROUTINE RDMATF (NS,NC,NOB,NG,ISAF,ISAG,ISAH,ISAD,IGAM,ISAA,ISAB
     1,IRDMAT,IFDFW)
       DATA IYES/'Y'/,INO/'N'/
C      INTEGER NS,NC,NOB,NG,ISAF,ISAG,ISAH,IGAM,IRDMAT,INO,IANS,K
       REWIND 9
       READ (9,240,END=30,ERR=30) K,IANS
       IF (IANS.EQ.1) GO TO 10
       GO TO 30
10     READ (9,250) NS,NC,NOB,NG,IFDFW
       WRITE (5,255)
       CALL FRTCMS ('CLRSCRN ')
20     WRITE (5,260)
       CALL RDINT (IANS)
       IF (IANS.GT.3) GO TO 20
       IF (IANS.EQ.3) GO TO 30
       IRDMAT=1
       IF (IANS.EQ.2) GO TO 40
       ISAF=1
       ISAG=1
       ISAH=1
       ISAD=1
       IGAM=1
       ISAA=1
       ISAB=1
30     RETURN
C--------------------------------------------ISAF-------------------
40     CALL FRTCMS ('CLRSCRN ')
```

```
440   FORMAT (5X,7HSTATE #,I3,2H =)
450   FORMAT (//,5X,52HDO YOU WISH TO CHANGE ANY OF THE SIGNIFICANT STAT
     1ES?,//,10X,19HTYPE "YES" OR "NO".)
460   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY...ENTER "YES" OR "NO".)
470   FORMAT (5X,37HENTER THE N-TH STATE # TO BE CHANGED.)
480   FORMAT (5X,17HENTER NEW STATE #,I3)
490   FORMAT (1X,63H***** WARNING: STATE NUMBERS HAVE BEEN IMPROPERLY EN
     1TERED *****/,16X,39HSTATE #'S MUST BE IN THE RANGE OF 1 AND,I3,/,2
     24X,25HAND IN ASCENDING ORDER!!!//)
500   FORMAT ()
510   FORMAT (I1,3X,I1)
      END
C=====================================================================
      SUBROUTINE MAMULT (L,M,N,F,G,H,NL,NM,NN)
C=====================================================================
C---------------------------------------------------------------------
      INTEGER I,J,K
      REAL*8 F(L,M),G(M,N),H(L,N),DEL
C---------------------------------------------------------------------
      DO 20 I=1,NL
      DO 20 K=1,NN
      DEL=0.D0
      DO 10 J=1,NM
10    DEL=DEL+F(I,J)*G(J,K)
20    H(I,K)=DEL
      RETURN
      END
C=====================================================================
      SUBROUTINE MI (M,N,AI,DI,DV,EV,FV,GV,HV)
C=====================================================================
C---------------------------------------------------------------------
      REAL*8 DI(M,M),DV(M),EV(M),AI(M,M),D,E,DABS
      INTEGER FV(M),GV(M),EF
      LOGICAL *1HV(M)
      COMMON ER,FMT
C---------------------------------------------------------------------
      DO 10 I=1,N
      HV(I)=.FALSE.
      DO 10 J=1,N
10    DI(I,J)=AI(I,J)
      DO 70 K=1,N
      L=0
      D=0.D0
      DO 20 J=1,N
      IF (HV(J)) GO TO 20
      E=DABS(DI(K,J))
      IF (E .LE. D) GO TO 20
      D=E
```

```
270     HR(I,J)=HR(I,J)-CSTR(I,J)
        WRITE (FMT,380)
        DO 280 I=1,NO
280     WRITE (FMT,320)  (HR(I,J),J=1,NR)
        IF (IFDFW.EQ.0) GO TO 310
        CALL MAMULT (NO,M,M,H2,F22,DUM7,NO,M,M)
        CALL MAMULT (NO,M,NC,DUM7,G2,DSTR,NO,M,NC)
        DO 290 I=1,NO
        DO 290 J=1,NC
290     DR(I,J)=D(I,J)-DSTR(I,J)
        WRITE (FMT,390)
        DO 300 I=1,NO
300     WRITE (FMT,320)  (DR(I,J),J=1,NC)
310     CONTINUE
        REWIND 10
        I=0
        IANS=1
        WRITE (10,510)  I,IANS
        WRITE (10,410)  NR,NC,NO,NG,IFDFW
        WRITE (10,400)  ((FR(I,J),J=1,NR),I=1,NR)
        WRITE (10,400)  ((GR(I,J),J=1,NC),I=1,NR)
        WRITE (10,400)  ((HR(I,J),J=1,NR),I=1,NO)
        WRITE (10,400)  ((DR(I,J),J=1,NC),I=1,NO)
        WRITE (10,400)  ((GAM(I,J),J=1,NG),I=1,NR)
        WRITE (10,400)  ((FBGC(I,J),J=1,NR),I=1,NC)
        WRITE (10,400)  ((FBGE(I,J),J=1,NO),I=1,NR)
        WRITE (10,400)  ((AY(I,J),J=1,NO),I=1,NO)
        WRITE (10,400)  ((B(I,J),J=1,NC),I=1,NC)
        RETURN
C-----------------------------------------------------------------------
C   55 FORMAT(2(0PD13.5))
C-----------------------------------------------------------------------
320     FORMAT (10(2X,1PD11.4))
330     FORMAT (13I5)
340     FORMAT (///,3X,36HTHE DESIRED REDUCED STATE ORDER IS:  ,I3)
350     FORMAT (//,3X,27HTHE SIGNIFICANT STATES ARE:)
360     FORMAT (//,3X,28HTHE REDUCED PLANT MATRIX IS:/)
370     FORMAT (//,3X,43HTHE REDUCED CONTROL DISTRIBUTION MATRIX IS:/)
380     FORMAT (//,3X,42HTHE REDUCED OUTPUT DISTRIBUTION MATRIX IS:/)
390     FORMAT (//,3X,34HTHE REDUCED FEEDFORWARD MATRIX IS:/)
400     FORMAT (4D20.13)
410     FORMAT (5I5)
420     FORMAT (5X,67HENTER THE "N" SIGNIFICANT STATES WHICH REPRESENT THE
       1 REDUCED ORDER /,5X,42HMODEL. ENTER STATE #'S IN ASCENDING ORDER.)
430     FORMAT (5X,58HDO YOU WISH TO INPUT DESIRED STATES FOR YOUR REDUCED
       1 ORDER/,5X,23HMODEL FROM A DATA FILE?///,5X,57HDATA FILE MUST BE N
       2AMED "STATES DATA A1" IN FIXED FORMAT./,5X,49HTHE READ FORMAT IS
       3"13I5" PER 72 CHARACTER LINE./,10X,19HTYPE "YES" OR "NO".)
```

```
      CALL MAMULT (M,M,NR,F22,F21,DUM2,M,M,NR)
      CALL MAMULT (NR,M,NR,F12,DUM2,FM,NR,M,NR)
      DO 190 I=1,NR
      DO 190 J=1,NR
190   FR(I,J)=F11(I,J)-FM(I,J)
      WRITE (FMT,360)
      DO 200 I=1,NR
200   WRITE (FMT,320) (FR(I,J),J=1,NR)
C-----------------------------------------------------------------------
C     COMPUTE THE REDUCED ORDER INPUT MATRIX.
C            G1   = < RM >*< G >
C            G2
C          < GR > = < G1 > - < F12 >*< F22**-1 >*< G2 >
C-----------------------------------------------------------------------
      CALL MAMULT (NS,NS,NC,RM,G,GM,NS,NS,NC)
      DO 210 I=1,NR
      DO 210 J=1,NC
210   GR(I,J)=GM(I,J)
      DO 220 I=1,M
      DO 220 J=1,NC
220   G2(I,J)=GM((I+NR),J)
      CALL MAMULT (NR,M,M,F12,F22,DUM3,NR,M,M)
      CALL MAMULT (NR,M,NC,DUM3,G2,DUM4,NR,M,NC)
      DO 230 I=1,NR
      DO 230 J=1,NC
230   GR(I,J)=GR(I,J)-DUM4(I,J)
      WRITE (FMT,370)
      DO 240 I=1,NR
240   WRITE (FMT,320) (GR(I,J),J=1,NC)
C-----------------------------------------------------------------------
C     COMPUTE THE REDUCED ORDER OUTPUT MATRICES.
C            < H1  H2 > = < H >*< RM**-1 >
C            < HR > = < H1 > - < H2 >*< F22**-1 >*< F21 >
C            < DR > = < D > - < H2 >*< F22**-1 >*< G2 >
C-----------------------------------------------------------------------
C     WRITE(6,983) NS,NC,NO,NR,M
      CALL MI (NS,NS,RM,DI,CV,EV,FV,GV,HV)
      CALL MAMULT (NO,NS,NS,H,RM,DUM5,NO,NS,NS)
      DO 250 I=1,NO
      DO 250 J=1,NR
250   HR(I,J)=DUM5(I,J)
      DO 260 I=1,NO
      DO 260 J=1,M
260   H2(I,J)=DUM5(I,(NR+J))
      CALL MAMULT (NO,M,M,H2,F22,HSTR,NO,M,M)
      CALL MAMULT (NO,M,NR,HSTR,F21,CSTR,NO,M,NR)
      DO 270 I=1,NO
      DO 270 J=1,NR
```

```
      DO 110 I=1,NS
      DO 110 J=1,NS
      RM(I,J)=0.D0
110   CONTINUE
      K=NR
      DO 140 I=1,NS
      DO 120 J=1,NR
      IF (I.EQ.NRS(J)) GO TC 130
120   CONTINUE
      K=K+1
      RM(K,I)=1.D0
      GO TO 140
130   RM(J,I)=1.D0
140   CONTINUE
C-----------------------------------------------------------------
C     DIVIDE THE PLANT MATRIX INTO THE FOUR SUBMATRICES
C     USED IN THE ORDER REDUCTION.            F11F12
C           <F'> = <RM>*<F>*<RM**-1> = < ------ >
C                                             F21F22
C-----------------------------------------------------------------
      CALL MAMULT (NS,NS,NS,RM,F,DUM1,NS,NS,NS)
      CALL MI (NS,NS,RM,DI,LV,EV,FV,GV,HV)
      CALL MAMULT (NS,NS,NS,DUM1,RM,T,NS,NS,NS)
      CALL MI (NS,NS,RM,DI,DV,EV,FV,GV,HV)
      DO 150 I=1,NR
      DO 150 J=1,NR
150   F11(I,J)=T(I,J)
      Q=NR+1
      M=NS-NR
      DO 160 I=1,NR
      DO 160 J=Q,NS
      P=J-NR
160   F12(I,P)=T(I,J)
      DO 170 I=Q,NS
      DO 170 J=1,NR
      P=I-NR
170   F21(P,J)=T(I,J)
      DO 180 I=Q,NS
      DO 180 J=Q,NS
180   F22((I-NR),(J-NR))=T(I,J)
C-----------------------------------------------------------------
C     COMPUTE THE REDUCED PIANT MATRIX.
C          < FR > = < F11 > - < F12 >*< F22**-1 >*< F21 >
C-----------------------------------------------------------------
      WRITE (FMT,340) NR
      WRITE (FMT,350)
      WRITE (FMT,330) (NRS(I),I=1,NR)
      CALL MI (M,M,F22,DI,DV,EV,FV,GV,HV)
```

```
C       FORM THE REORDERED STATE VECTOR X'
C
C               < X' > = < RM > * < X >
C-----------------------------------------------------------------------
10      CONTINUE
        CALL FRTCMS ('CLRSCRN ')
        WRITE (5,430)
        CALL RDCHAR (IANS)
        CALL FRTCMS ('CLRSCRN ')
        IF (IANS.EQ.IY) ISTA=1
        IF (ISTA.EQ.1) GO TO 90
20      WRITE (5,420)
        DO 40 I=1,NR
        WRITE (5,440) I
        CALL RDREAL (ANSR)
        NRS(I)=IDINT(ANSR)
        IF ((NRS(I).LE.0).OR.(NRS(I).GT.NS)) GO TO 30
        IF (I.EQ.1) GO TO 40
        IF (NRS(I).GT.NRS(I-1)) GO TO 40
30      WRITE (5,490) NS
        GO TO 20
40      CONTINUE
        CALL FRTCMS ('CLRSCRN ')
        WRITE (5,350)
        WRITE (5,330) (NRS(I),I=1,NR)
50      CONTINUE
        WRITE (5,450)
        CALL RDCHAR (IANS)
        CALL FRTCMS ('CLRSCRN ')
        IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
        GO TO 70
60      WRITE (5,460)
        GO TO 50
70      CONTINU
        IF (IANS.EQ.IZ) GO TO 100
        IF (IANS.EQ.IY) GO TO 80
80      WRITE (5,470)
        CALL RDREAL (ANSR)
        DUM=IDINT(ANSR)
        WRITE (5,480) DUM
        CALL RDREAL (ANSR)
        K=IDINT(ANSR)
        NRS(DUM)=K
        GO TO 40
90      READ (7,330) (NRS(I),I=1,NR)
        GO TO 40
100     CONTINUE
        CALL FRTCMS ('CLRSCRN ')
```

```
     8NUE?//,10X,19HTYPE "YES" OR "NO".)
290   FORMAT (1X,64H*** WARNING: REDUCED ORDER MUST BE GREATER THAN 0 AN
     1D LESS THAN ,I3,4H ***//)
300   FORMAT (5X,38HDO YOU WISH RESULTS TO SCREEN OR DISK?//,04X,52HNOTE
     1: A DATA FILE CONTAINING THE REDUCED ORDER DATA /,10X,44HWILL BE G
     2ENERATED REGARDLESS OF YOUR ANSWER.//,10X,36HTYPE "S" FOR SCREEN O
     3R "D" FOR DISK.)
310   FORMAT (//,19X,26HDO YOU DESIRE ANOTHER RUN?//,14X,36HIF YES, THE
     1RESULTS OF THIS RUN WILL/,17X,31HREPLACE YOUR LAST OPTMATR DATA.//
     2,23X,19HTYPE "YES" OR "NO".)
      END
C=======================================================================
      SUBROUTINE REDUCX (NG,IFDPW,M,NS,NC,NO,NR,F,G,H,D,FR,GR,HR,DR
     1 ,NRS,F11,F12,F21,F22,GM,H2,G2,T,HSTR,FM,DSTR,DUM1
     2 ,DUM3,DUM4,DUM2,CSTR,RM,DUM5,DUM7,AI,DI,DV,EV,FV,GV,HV
     3 ,GAM,FBGC,FBGE,AY,B)
C=======================================================================
      INTEGER I,J,K,L,M,Q,DUM,IANS,IFDPW,NRS(NR),FV(NS),GV(NS),NG,I
     1STA,FMT
      LOGICAL *1HV(NS)
      REAL*8 F(NS,NS),G(NS,NC),H(NO,NS),D(NO,NC),GAM(NS,NG),FBGC(NC,NS)
     1,F11(NR,NR),F12(NR,M),F21(M,NR),F22(M,M),FBGE(NS,NO),AY(NO,NO)
     2,GM(NS,NC),H2(NO,M),G2(M,NC),T(NS,NS),B(NC,NC)
     3,FR(NR,NR),HR(NO,NR),DR(NO,NC),HSTR(NO,M),FM(NR,NR)
     4,DSTR(NO,NC),GR(NR,NC),DV(NS),EV(NS)
     5,DUM1(NS,NS),DUM3(NR,M),DUM4(NR,NC),DUM2(M,NR),CSTR(NO,NR)
     6,RM(NS,NS),DUM5(NO,NS),DUM7(NO,M),ANSR,AI(NS,NS),DI(NS,NS)
      DATA IY/'Y'/,IZ/'N'/
      COMMON ER,FMT
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C    READ IN THE FULL ORDER MODEL MATRICES FROM OPTMAT DATA FILE
C-----------------------------------------------------------------------
      ISTA=0
      REWIND 9
      READ (9,500)
      READ (9,500)
      READ (9,400) ((F(I,J),J=1,NS),I=1,NS)
      READ (9,400) ((G(I,J),J=1,NC),I=1,NS)
      READ (9,400) ((H(I,J),J=1,NS),I=1,NO)
      READ (9,400) ((D(I,J),J=1,NC),I=1,NO)
      READ (9,400) ((GAM(I,J),J=1,NG),I=1,NS)
      READ (9,400) ((FBGC(I,J),J=1,NS),I=1,NC)
      READ (9,400) ((FBGE(I,J),J=1,NO),I=1,NS)
      READ (9,400) ((AY(I,J),J=1,NO),I=1,NO)
      READ (9,400) ((B(I,J),J=1,NC),I=1,NC)
C-----------------------------------------------------------------------
C     READ IN THE DESIRED REDUCED ORDER AND THE DESIRED STATES.
```

```
      GO TO 110
130   M=NS-NR
C-----------------------------------------------------------------------
C   COMPUTE REDUCED ORDER F, G, H, AND D MATRICES
C-----------------------------------------------------------------------
      CALL REDUCX (NG,IFDFW,M,NS,NC,NO,NR,F,G,H,D,FR,GR,HR,DR
     1,NRS,F11,F12,F21,F22,GM,H2,G2,T,HSTR,FM,DSTR,DUM1
     2,DUM3,DUM4,DUM2,CSTR,RM,DUM5,DUM7,AI,DI,DV,EV,FV,GV,HV
     3,GAM,FBGC,FBGE,AY,B)
      WRITE (5,260)
      WRITE (5,250)
      IF (ER.EQ.0) GO TO 150
      WRITE (5,270)
      GO TO 150
140   WRITE (5,230)
      GO TO 190
150   WRITE (5,310)
160   CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 170
      GO TO 180
170   WRITE (5,220)
      GO TO 160
180   CONTINUE
      IF (IANS.EQ.IY) GO TO 100
      IF (IANS.EQ.IZ) GO TO 140
190   CONTINUE
      STOP
200   FORMAT (5I5)
210   FORMAT (5X,33HTHE ORDER OF THE FULL SYSTEM IS: ,I3,/,5X,26HTHE NUM
     1BER OF CONTROLS IS: ,I3,/,5X,31HTHE NUMBER OF OBSERVATIONS IS: ,I3,
     2/,5X,18HA "D" MATRIX WILL ,A3,10H BE INPUT.//,5X,24HDO YOU WISH TO
     3 CONTINUE?,//,10X,19HTYPE "YES" OR "NO".)
220   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY...ENTER "YES" OR "NO".)
230   FORMAT (//,13X,40H........OPTRED IS NOW TERMINATED........//)
240   FORMAT (5X,50HENTER THE DESIRED REDUCED ORDER OF THE "F" MATRIX.)
250   FORMAT (5X,59HANALYSIS COMPLETE...YOUR REDUCED SYSTEM DATA HAS BEE
     1N SAVED/,18X,31HIN A FILE NAMED "OPTMATR DATA".)
260   FORMAT (///)
270   FORMAT (20X,22H******** NOTE ********//,10X,42HTHE RESULTS OF THIS
     1 RUN WILL BE INVALID!!!)
280   FORMAT (5X,59HOPTRED WILL COMPUTE A REDUCED ORDER MODEL FROM FULL
     1SYSTEM /,5X,41H"F","G","H", AND "D" (IF INPUT) MATRICES.//,5X,56HT
     2HE FULL SYSTEM MATRICES MUST BE READ FROM A  FILE NAMED/,5X,48H"OP
     3TMAT DATA" AS CREATED BY THE OPTSYSX PROGRAM.//,5X,64HYOU MUST ALS
     4O ENTER THE DESIRED REDUCED ORDER (NUMBER OF STATES)/,5X,62HAND TH
     5E ACTUAL STATE #'S  (IN ASCENDING ORDER) WHICH REPRESENT/,5X,58HTH
     6E REDUCED MODEL. THE ORDER OF THE REDUCED MODEL MUST BE /,5X,39HLE
     7SS THAN THE ORDER OF THE FULL SYSTEM.//,5X,24HDO YOU WISH TO CONTI
```

```
C-----------------------------------------------------------------------
C  SUPPRESS INDIVIDUAL UNDER/OVER FLOW ERROR MESSAGES; PROVIDE SUMMARY
C-----------------------------------------------------------------------
      CALL ERRSET (207,256,-1,1,1,209)
C----------------------PROGRAM DESCRIPTION------------------------------
      CALL FRTCMS ('CLRSCRN ')
10    WRITE (5,280)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 20
      GO TO 30
20    WRITE (5,220)
      GO TO 10
30    IF (IANS.EQ.IZ) GO TO 140
      CALL FRTCMS ('CLRSCRN ')
40    WRITE (5,300)
      CALL RDCHAR (IANS)
      IF ((IANS.NE.SC).AND.(IANS.NE.DK)) GO TO 50
      GO TO 60
50    WRITE (5,220)
      GO TO 40
60    IF (IANS.EQ.SC) FMT=5
      IF (IANS.EQ.DK) FMT=6
C-----------------------------------------------------------------------
C  READ SYSTEM FLAGS, FULL SYSTEM MATRIX PARAMETERS FROM OPTMAT DATA
C  AND READ DESIRED REDUCED ORDER
C-----------------------------------------------------------------------
      CALL FRTCMS ('CLRSCRN ')
      ER=0
      DD=YY
      READ (9,200) K,IANS
      READ (9,200) NS,NC,NO,NG,IFDFW
      IF (IFDFW.EQ.0) DD=NN
      WRITE (5,210) NS,NC,NC,DD
70    CALL RDCHAR (IANS)
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 80
      GO TO 90
80    WRITE (5,220)
      GO TO 70
90    CONTINUE
      IF (IANS.EQ.IZ) GO TO 140
      IF (IANS.EQ.IY) GO TO 100
100   CALL FRTCMS ('CLRSCRN ')
110   WRITE (5,240)
      CALL RDREAL (ANSR)
      NR=IDINT(ANSR)
      IF ((NR.GE.NS).OR.(NR.LE.0)) GO TO 120
      GO TO 130
120   WRITE (5,290) NS
```

APPENDIX B

THE OPTRED PROGRAM

```
C***********************************************************************
C***********************************************************************
C*********************      OPTRED FORTRAN      ************************
C****************                                    *******************
C  A SELF CONTAINED, FULLY INTERACTIVE PROGRAM WHICH READS THE FULL   **
C***   SYSTEM "F","G","H", AND "D" MATRICES FROM THE DATA FILE    *******
C****    "OPTMAT" CREATED BY THE OPTSYSX PROGRAM. ITS OUTPUT      ********
C*******     IS REDUCED ORDER "F","G","H", AND "D" MATRICES    **********
C*************      FORMATTED FOR RE-ANALYSIS BY OPTSYSX    *************
C*************                                              *************
C*********************      BY STAN NELSON      ************************
C*********************      DECEMBER 1984       ************************
C***********************************************************************
C----------------------NORMAL DIMENSIONS--------------------------------
C
C      REAL*8 F(16,16),G(16,8),H(8,16),D(8,8),RM(16,16)
C     1,F11(8,8),F12(8,16),F21(16,8),F22(16,16)
C     2,H2(8,16),G2(16,8),T(16,16),GAM(16,1),FBGC(8,16),FBGE(16,8)
C     3,FR(8,8),GR(8,8),HR(8,8),DR(8,8),HSTR(8,16),FM(8,8),DUM1(16,16)
C     4,DUM3(8,16),DUM4(8,8),DUM2(16,8),CSTR(8,8),GAM(16,1),AY(8,8)
C     5,DUM5(8,16),DUM7(8,16),GM(16,8),DSTR(8,8),B(8,8)
C     6,DV(16),EV(16),DI(16,16),AI(16,16),ANSR
C      INTEGER IFDFW,M,N,NS,NC,NO,NR,I,J,NRS(5),FV(16),GV(16),NG,K,IANS
C     1,ER,DD,YY,NN,SC,DK,FMT
C      COMMON ER,FMT
C      LOGICAL*1 HV(16)
C
C----------------------INCREASED DIMENSIONS-----------------------------
C
      REAL*8 F(98,98),G(98,3),H(8,98),D(8,8),RM(98,98)
     1,F11(60,60),F12(60,60),F21(60,60),F22(60,60)
     2,H2(8,60),G2(60,8),T(98,98),GAM(98,1),FBGC(8,98),FBGE(98,8)
     3,FR(60,60),GR(60,8),HR(8,60),DR(8,8),HSTR(8,60),FM(60,60)
     4,DUM3(60,60),DUM4(60,8),DUM2(60,60),CSTR(8,60),AY(8,8),B(8,8)
     5,DUM5(8,98),DUM7(8,60),GM(98,8),DSTR(8,8),DUM1(98,98)
     6,DV(98),EV(98),DI(98,98),AI(98,98),ANSR
      INTEGER IFDFW,M,N,NS,NC,NO,NR,I,J,NRS(60),FV(98),GV(98),NG,K,IANS,
     1DD,NN,YY,ER,SC,DK,FMT
      LOGICAL *1HV(98)
      COMMON ER,FMT
      DATA IY/'Y'/,IZ/'N'/,NN/3HNOT/,YY/3H   /,SC/'S'/,DK/'D'/
```

```
     212X,'PLOT,  OR NICHOLS PLOT OF THE SYSTEM YOU ARE EVALUATING?',
     3/,24X,'(Y OR N)'
     4///,5X,'NOTE: YOU MUST BE LOGGED ON AT A DUAL SCREEN',/,
     410X,'(TEK 618) TERMINAL TO UTILIZE THIS MODE.',///,10X,
     558HTHE F (SYSTEM), G (CONTROL), H (OBSERVABLES), GAM (NOISE),,/,
     613X,53HA (OUTPUT COST) AND B (CONTROL COST) MATRICES WILL BE,/,
     716X,'SAVED FOR REENTRY TO THE MAIN OPTSYS PROGRAM.')
110   FORMAT(10X,29HYOU MUST ANSWER (Y)ES OR (N)O  )
120   FORMAT(5I5)
130   FORMAT(4D20.13)
140   FORMAT(I1,3X,I1)
      END
```

```
      READ(9,10)  ((D(I,J),J=1,NCI),I=1,NOI)
      READ(9,10)  ((GAM(I,J),J=1,NGI),I=1,NSI)
      READ(9,10)  ((FBGC(I,J),J=1,NSI),I=1,NCI)
      READ(9,10)  ((FBGE(I,J),J=1,NOI),I=1,NSI)
      READ(9,10)  ((AY(I,J),J=1,NOI),I=1,NOI)
      READ(9,10)  ((B(I,J),J=1,NCI),I=1,NCI)
      RETURN
C-----------------------------------------------------------------------
10    FORMAT(4(D20.13))
20    FORMAT(5I5)
      END
C========================================================================
C  SUBROUTINE WRTMAT -- WRITES THE F, G, HO & GAM MATRICES TO           =
C  THE DATA FILE OPTMAT ON FILEDEF 9.                                   =
C========================================================================
      SUBROUTINE WRTMAT(BA,G,HO,D,GAM,FBGC,FBGE,AY,B,NS,NC,NO,NG,IFDFW)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BA(NS,NS),G(NS,NC),HO(NO,NS),GAM(NS,NG),FBGC(NC,NS),
     1AY(NO,NO),B(NC,NC),FBGE(NS,NO),D(NO,NC)
C     INTEGER NS,NC,NO,NG,I,J,IANS,INO,IYES,IFDFW
      DATA IYES/'Y'/,INO/'N'/
      WRITE (5,90)
      CALL FRTCMS('CLRSCRN ')
10    WRITE(5,100)
      CALL RDCHAR(IANS)
      IF((IANS.EQ.INO).OR.(IANS.EQ.IYES)) GO TO 20
      WRITE(5,110)
      GO TO 10
20    IF (IANS.EQ.INO) RETURN
      REWIND 9
      I = 0
      IANS = 1
      WRITE(9,140) I,IANS
      WRITE(9,120) NS,NC,NO,NG,IFDFW
      WRITE(9,130)  ((BA(I,J),J=1,NS),I=1,NS)
      WRITE(9,130)  ((G(I,J),J=1,NC),I=1,NS)
      WRITE(9,130)  ((HO(I,J),J=1,NS),I=1,NO)
      WRITE(9,130)  ((D(I,J),J=1,NC),I=1,NO)
      WRITE(9,130)  ((GAM(I,J),J=1,NG),I=1,NS)
      WRITE(9,130)  ((FBGC(I,J),J=1,NS),I=1,NC)
      WRITE(9,130)  ((FBGE(I,J),J=1,NO),I=1,NS)
      WRITE(9,130)  ((AY(I,J),J=1,NO),I=1,NO)
      WRITE(9,130)  ((B(I,J),J=1,NC),I=1,NC)
      STOP
C-----------------------------------------------------------------------
90    FORMAT(//////////////////////)
100   FORMAT(///,10X,'DO YOU WISH TO OBTAIN A TIME RESPONSE,',/,
     112X,'POLE-ZERO MAP, RCOT-LOCUS PLOT,  BODE PLOT,  NYQUIST',/,
```

```
280   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "H"-MATRIX FROM THE LAST
     1,/,5X,   'RUN TO BE USED IN THIS RUN?'          ,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
285   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "D"-MATRIX FROM THE LAST
     1,/,5X,   'RUN TO BE USED IN THIS RUN?'          ,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
290   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "G"-MATRIX FROM THE LAST
     1,/,5X,   'RUN TO BE USED IN THIS RUN?'          ,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
300   FORMAT (///,5X,52HDO YOU WISH TO SAVE THE "GAMMA"-MATRIX FROM THE
     1LAST,/,5X,   'RUN TO BE USED IN THIS RUN?'          ,//,5X,39HNOTE: T
     2HE MATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE
     3 INTERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27
     4HINDIVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
310   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "A"-MATRIX FROM THE LAST
     1,/,5X,   'RUN TO BE USED IN THIS RUN?'          ,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
320   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "B"-MATRIX FROM THE LAST
     1,/,5X,   'RUN TO BE USED IN THIS RUN?'          ,//,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
330   FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY!  ENTER "YES" OR "NO".)
      END
C======================================================================
C   SUBROUTINE RDMAT--READS THE F, G, H, D, GAM, A AND B MATRICES FROM  =
C   MATRICES FROM THE DATA FILE OPTMAT ON FILEDEF 9.                    =
C======================================================================
      SUBROUTINE RDMAT(BA,G,HO,D,GAM,FBGC,FBGE,AY,B,NS,NC,NO,NG,IRDMAT,
     1IFDFW)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION BA(NS,NS),G(NS,NC),HO(NO,NS),GAM(NS,NG),FBGC(NC,NS),
     2AY(NO,NO),B(NC,NC),FBGE(NS,NO),D(NO,NC)
      IF(IRDMAT.EQ.0) RETURN
      REWIND 9
      READ (9,20) K,IANS
      READ (9,20) NSI,NCI,NCI,NGI,IFDFW
      READ(9,10)  ((BA(I,J),J=1,NSI),I=1,NSI)
      READ(9,10)  ((G(I,J),J=1,NCI),I=1,NSI)
      READ(9,10)  ((HO(I,J),J=1,NSI),I=1,NOI)
```

```
160   WRITE (5,300)
      CALL RDCHAR (IANS)
      IF ((IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 180
170   WRITE (5,330)
      GO TO 160
180   CONTINUE
      IF (IANS.EQ.IYES) IGAM=1
      IF (IANS.EQ.INO) IGAM=0
190   CONTINUE
C-------------------------------------------ISAA------------------
      CALL FRTCMS ('CLRSCRN ')
200   WRITE (5,310)
      CALL RDCHAR (IANS)
      IF ((IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 210
      WRITE (5,330)
      GO TO 200
210   CONTINUE
      IF (IANS.EQ.IYES) ISAA=1
      IF (IANS.EQ.INO) ISAA=0
C-------------------------------------------ISAB------------------
      CALL FRTCMS ('CLRSCRN ')
220   WRITE (5,320)
      CALL RDCHAR (IANS)
      IF ((IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 230
      WRITE (5,330)
      GO TO 220
230   CONTINUE
      IF (IANS.EQ.IYES) ISAB=1
      IF (IANS.EQ.INO) ISAB=0
      RETURN
C-----------------------------------------------------------------
240   FORMAT (I1,3X,I1)
250   FORMAT (5I5)
255   FORMAT (////////////////////)
260   FORMAT (//,10X,47HTHE "F","G","H","D","GAM", "A" AND "B" MATRICES,
     1/,12X,42H FROM YOUR PREVIOUS OPTSYS RUN WERE SAVED.,//,10X,36HTHE
     2FOLLOWING OPTIONS ARE AVAILABLE:,/15X,38H1. USE ALL OF THE SAME MA
     3TRICES AGAIN.,/,15X,'2. USE SELECTED MATRICES AGAIN.',/,
     415X,'3. INPUT ALL NEW MATRICES.',//,10X,17HENTER 1, 2, OR 3.
     5,//,10X,'NOTE: EACH SAVED MATRIX WILL BE REDISPLAYED AT'
     6,/,10X,34HTHE PROPER INPUT SEQUENCE INTERVAL
     7,/,10X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,10X,
     827HINDIVIDUAL MATRIX ELEMENTS.)
270   FORMAT (///,5X,48HDO YOU WISH TO SAVE THE "F"-MATRIX FROM THE LAST
     1,/,5X,  'RUN TO BE USED IN THIS RUN?'        //,5X,39HNOTE: THE M
     2ATRIX WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE INT
     3ERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HIND
     4IVIDUAL MATRIX ELEMENTS.,//,15X,19HTYPE "YES" OR "NO".)
```

```
50     WRITE (5,270)
       CALL RDCHAR (IANS)
       IF ((IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 70
60     WRITE (5,330)
       GO TO 50
70     CONTINUE
       IF (IANS.EQ.IYES) ISAF=1
       IF (IANS.EQ.INO) ISAF=0
C----------------------------------------------ISAH------------------
       IF (NOB.EQ.0) GO TO 110
       CALL FRTCMS ('CLRSCRN ')
80     WRITE (5,280)
       CALL RDCHAR (IANS)
       IF ((IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 100
90     WRITE (5,330)
       GO TO 80
100    CONTINUE
       IF (IANS.EQ.IYES) ISAH=1
       IF (IANS.EQ.INO) ISAH=0
110    CONTINUE
C----------------------------------------------ISAG------------------
       IF (NC.EQ.0) GO TO 150
       CALL FRTCMS ('CLRSCRN ')
120    WRITE (5,290)
       CALL RDCHAR (IANS)
       IF ((IANS.EQ.IYES).OR.(IANS.EQ.INO)) GO TO 140
130    WRITE (5,330)
       GO TO 120
140    CONTINUE
       IF (IANS.EQ.IYES) ISAG=1
       IF (IANS.EQ.INO) ISAG=0
150    CONTINUE
C----------------------------------------------ISAD------------------
       IF (IFDFW.EQ.0) GO TO 155
       CALL FRTCMS ('CLRSCRN ')
151    WRITE (5,285)
       CALL RDCHAR (IANS)
       IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 153
       GO TO 154
153    WRITE (5,330)
       GO TO 151
154    CONTINUE
       IF (IANS.EQ.IY) ISAD=1
       IF (IANS.EQ.IZ) ISAD=0
155    CONTINUE
C----------------------------------------------IGAM------------------
       IF (NG.EQ.0) GO TO 190
       CALL FRTCMS ('CLRSCRN ')
```

```
      L=J
20    CONTINUE
      IF (L) 30,30,40
30    IF (ER.EQ.1) RETURN
      WRITE (6,90)
      WRITE (5,90)
      ER=1
      RETURN
40    D=1.D0/DI(K,L)
      DO 50 I=1,N
      DV(I)=D*DI(I,L)
      EV(I)=DI(K,I)
      DI(I,L)=0.D0
50    DI(K,I)=0.D0
      DV(K)=D
      EV(L)=-1.D0
      DO 60 I=1,N
      DO 60 J=1,N
60    DI(I,J)=DI(I,J)-DV(I)*EV(J)
      DI(K,L)=D
      FV(K)=L
      GV(L)=K
70    HV(L)=.TRUE.
      DO 80 I=1,N
      K=FV(I)
      DO 80 J=1,N
      L=GV(J)
80    AI(K,L)=DI(I,J)
      RETURN
C-----------------------------------------------------------------------
90    FORMAT (//,5X,60HOPTRED CANNOT COMPUTE A REDUCED ORDER MODEL USING
     1 THE STATES/,5X,19HYOU HAVE REQUESTED.//,5X,62HENSURE THAT YOU HAV
     2E ENTERED THE SIGNIFICANT STATES CORRECTLY./,5X,66HOTHERWISE, A DI
     3FFERENT SET OF SIGNIFICANT STATES MUST BE SELECTED.////,20X,22H***
     4***** NOTE ********//,10X,42HTHE RESULTS OF THIS RUN WILL BE INVAL
     5ID!!!)
      END
C=======================================================================
C   SUBROUTINE RDREAL -- INTERACTIVELY READS A REAL NUMBER REPLY       =
C   INTO A FORTRAN PROGRAM.  IF THE USER INADVERTENLY ENTERS  A NULL   =
C   STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.             =
C=======================================================================
      SUBROUTINE RDREAL (ANSR)
      REAL*8 ANSR
      INTEGER COUNT
C-----------------------------------------------------------------------
      COUNT=0
10    CONTINUE
```

```
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      READ (5,*,END=30,ERR=30) ANSR
      RETURN
30    REWIND 5
      WRITE (5,50)
      GO TO 10
40    CONTINUE
      STOP
C-----------------------------------------------------------------------
50    FORMAT (1X,64HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER A NUME
     1RICAL VALUE.)
60    FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED!)
      END
C=======================================================================
C   SUBROUTINE RDINT -- INTERACTIVELY READS AN INTEGER REPLY           =
C   INTO A FORTRAN PROGRAM.  IF THE USER INADVERTENLY ENTERS AN IMPROPER=
C   DATA CHARACTER THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.     =
C=======================================================================
      SUBROUTINE RDINT (IANS)
      INTEGER COUNT,IANS
C-----------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 50
20    CONTINUE
      READ (5,*,END=40,ERR=40) IANS
      IF (IANS) 40,40,30
30    CONTINUE
      RETURN
40    REWIND 5
      WRITE (5,70)
      GO TO 10
50    CONTINUE
      STOP
C-----------------------------------------------------------------------
60    FORMAT (//,5X,49HPROGRAM TERMINATION - TWO IMPROPER DATA ENTRIES!!
     1)
70    FORMAT (1X,56HWARNING: IMPROPER DATA ENTRY!  ENTER A POSITIVE INTE
     1GER.)
      END
C=======================================================================
```

```
C  SUBROUTINE RDCHAR -- INTERACTIVELY READS A CHARACTER STRING REPLY   =
C  ('YES' OR 'NO') INTO A FCRTRAN PROGRAM. IF THE USER INADVERTENLY    =
C  ENTERS  A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY=
C======================================================================
      SUBROUTINE RDCHAR (IANS)
      INTEGER COUNT,IANS
C     DATA IY/'Y'/,IZ/'N'/
C----------------------------------------------------------------------
      COUNT=0
10    CONTINUE
      COUNT=COUNT+1
      IF (COUNT.LT.3) GO TO 20
      WRITE (5,60)
      GO TO 40
20    CONTINUE
      REWIND 5
      READ (5,70,END=30,ERR=30) IANS
      RETURN
30    REWIND 5
      WRITE (5,50)
      GO TO 10
40    CONTINUE
      STOP
C----------------------------------------------------------------------
50    FORMAT (1X,60HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER "YES"
     1OR "NO".)
60    FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULI STRINGS ENTERED!)
70    FORMAT (A1)
      END
```

APPENDIX C

THE OPTRED EXEC

```
*====================================
*  THIS EXEC EXECUTES OPTRED PROGRAM  *
*====================================
FILEDEF 05 TERM
FILEDEF 06 DISK OPTRED LISTING A1
FILEDEF 07 DISK STATES DATA A1
FILEDEF 09 DISK OPTMAT DATA A1
FILEDEF 10 DISK OPTMATR DATA A1
GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLDP NONIMSL
LOAD OPTRED (START
```

LIST OF REFERENCES

1. Hall, W.E., Computational Methods for the Synthesis of Rotary- Wing VTOL Aircraft Control Systems, Ph.D. Dissertation, Stanford University, Palo Alto, California, 1971.

2. Stanford University, Aero/Astro Department, User's Manual for OPTSYS 4 at SCIP, by R.A. Walker, December 1979.

3. Stanford University, Aero/Astro Department, User's Manual for OPTSYS 5 at CIT, by G. Liu, August, 1982.

4. Hoden, J.G., Interactive Implementation of the Optimal Systems Control Design Program (OPTSYSX) on the IBM/3033, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1984.

5. Diel, H.A., Development of Graphical Time Response using the OPTSYSX Program, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1984.

6. Laptas, M.H., Development of Graphical Pole-Zero, Root-locus, Bode, Nyquist, and Nichols Responses using the OPTSYSX Program, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1984.

7. Bryson, A.E. and Ho, Y. C., Applied Optimal Control, Hemisphere Pub. Co., 1969, (2nd Printing, 1975).

8. DeHoff, R.L. and Hall, W.E., "State Space Techniques and Modeling for Control," Control and Dynamics, vol. 14, Academic Press, 1978.

9. Dougherty, B.I., Effect of Fuel Efficiency of Parameter Variations in the Cost Function for Multivariable Control of a Turbofan Engine, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1981.

10. Enns, D., Model Reduction for Control System Design, Ph.D. Dissertation, Stanford University, Aero/Astro Department, 1984.

BIBLIOGRAPHY

Kwakernaak, H., and Sivan, S., *Linear Optimal Control Systems*, Wiley-Interscience, 1972.

Leondes, C.T., Ed., *Control and Dynamic Systems*, vol. 14, Academic Press, 1978

Porter, B., and Crossley, T.R., *Modal Control*, Taylor & Francis LTD, 1972

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93943	1
3.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
4.	Professor D.J. Collins Code 67Co Department of Aeronautics Naval Postgraduate School Monterey, California 93943	5
5.	Professor H.A. Titus Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	1
6.	Mr. Joe Gera NASA Dryden Flight Research Center P.O. Box 273 Mail Code OFDC Edwards, California 93523	1
7.	LCDR S.W. Nelson 14031 Kimberley Lane Houston, Texas 77079	4

END

FILMED

7-85

DTIC